

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

CHYTRÁ SIGNALIZAČNÍ JEDNOTKA

SMART SIGNALING UNIT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

David Stejskal

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Ondřej Baštán

BRNO 2021

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: David Stejskal

ID: 211177

Ročník: 3

Akademický rok: 2020/21

NÁZEV TÉMATU:

Chytrá signalizační jednotka

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvořit chytrou signalizační jednotku signalizující stavy dle standardu pack-ML.

1. Seznamte se s dostupnými signalizačními jednotkami
2. Definujte požadavky na signalizační jednotku
3. Navrhněte koncepci zařízení
4. Sestavte HW zařízení
5. Navrhněte koncepci SW
6. Implementujte SW
7. Zařízení otestujte a zhodnoťte dosažené výsledky

DOPORUČENÁ LITERATURA:

GAJSKI, Daniel D., et al. Embedded system design: modeling, synthesis and verification. Springer Science & Business Media, 2009.

Termín zadání: 8.2.2021

Termín odevzdání: 24.5.2021

Vedoucí práce: Ing. Ondřej Baštán

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Podstatou této bakalářské práce je kompletní návrh hardwarového i programového řešení a následná realizace **Chytré signalizační jednotky**. Ta má nahradit aktuálně nainstalované signalizační **majáčky** a svými vlastnostmi zdokonalit stávající testbed samo-činného barmana.

Navržené zařízení je schopné **signalizovat stavy stroje podle standardu PackML**. Dále umí **kommunikovat po Ethernetu pomocí průmyslového komunikačního protokolu Modbus TCP**, přičemž realizuje Modbus server, který zajišťuje služby pro další připojená zařízení (Modbus klienty). Zařízení dále **implementuje standard Power over Ethernet (PoE)**, díky kterému je možné napájet signalizační jednotku přímo po ethernetovém kabelu, na němž zároveň probíhá i komunikace.

Na úvod jsou představeny zásadní cíle práce společně s požadavky na Chytrou signalizační jednotku. Následně je vysvětlen význam signalizačních jednotek v průmyslu a dále jsou uvedeny základní parametry pro výběr konkrétního produktu. V návaznosti na to byl proveden průzkum dostupných signálních věží na trhu.

V teoretické části práce je nejprve podrobněji popsána iniciativa Průmysl 4.0, na jejímž základě je stručně představen testbed samo-činný barman. Následuje obsáhlý popis průmyslových standardů PoE, Modbus TCP a PackML použitých v rámci návrhu zařízení. Koncepte signalizační jednotky spadá již do praktické části práce a zabývá se způsoby zobrazování, které představují primární funkci Chytré signalizační jednotky. Nedílnou součástí je koncepce hardwaru, která popisuje nejpodstatnější použité komponenty. Závěrečná část práce se zabývá tvorbou desky plošných spojů a vývojem firmwaru pomocí vývojového prostředí Visual Studio Code za použití frameworku ESP-IDF od firmy Espressif.

KLÍČOVÁ SLOVA

Průmysl 4.0, testbed samo-činný barman, Ethernet, Power over Ethernet, Modbus TCP, OMAC PackML, ESP32-WROOM-32, LED pásek WS2812B

ABSTRACT

The essence of this bachelor thesis is a complete design of hardware and software solutions and subsequent implementation of a **Smart Signaling Unit**. It is intended to replace the currently installed signaling **beacons** and to improve the existing testbed Self-Acting Barman with its features.

The designed device is able to **signal machine states according to the PackML standard**. It can also **communicate via Ethernet using the industrial communication protocol Modbus TCP**, while implementing a Modbus server, which provides services for other connected devices (Modbus clients). The device also implements the **Power over Ethernet (PoE)** standard, thanks to which it is possible to power the signaling unit directly via an Ethernet cable, on which communication also takes place. At the beginning, the main goals of the work are presented together with the requirements for the Smart Signaling Unit. Subsequently, the importance of signaling units in the industry is explained and the basic parameters for the selection of a specific product are given below. Following this, a survey of available signal towers on the market was carried out.

In the theoretical part of the work, the Industry 4.0 initiative is first described in more detail, on the basis of which the testbed Self-Acting Barman is briefly introduced. The following is a comprehensive description of the industry standards PoE, Modbus TCP and PackML used in device design.

The concept of the signaling unit already belongs to the practical part of the thesis and deals with the methods of display, which represent the primary function of the Smart Signaling Unit. An integral part is the hardware concept, which describes the most important used components.

The final part of the work deals with the creation of a printed circuit board and the development of firmware using the Visual Studio Code development environment and the framework ESP-IDF from the company Espressif.

KEYWORDS

Industry 4.0, Testbed Self-Acting Barman, Ethernet, Power over Ethernet, Modbus TCP, OMAC PackML, ESP32-WROOM-32, LED Strip WS2812B

STEJSKAL, David. *Chytrá signalizační jednotka*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2021, 71 s. Bakalářská práce. Vedoucí práce: Ing. Ondřej Baštán

Prohlášení autora o původnosti díla

Jméno a příjmení autora: David Stejskal
VUT ID autora: 211177
Typ práce: Bakalářská práce
Akademický rok: 2020/21
Téma závěrečné práce: Chytrá signalizační jednotka

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno
.....
podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Ondřeji Baštánovi za odborné vedení, konzultace, trpělivost a podnětné návrhy. Dále patří velké poděkování mé přítelkyni Bc. Petře Vopařilové za její trpělivost a psychickou podporu při psaní této závěrečné práce.

Obsah

Úvod	12
1 Cíle práce	13
1.1 Požadavky na Chytrou signalizační jednotku	13
1.1.1 Signalizace stavů	13
1.1.2 Komunikace přes Modbus TCP	14
1.1.3 Napájení přes Power over Ethernet	14
1.1.4 Zajištění teplotní stability	14
1.1.5 Konstrukce	14
2 Signalizační jednotka	15
2.1 Základní parametry	15
2.2 Průzkum trhu	16
2.3 Srovnání vybraných produktů	17
2.3.1 Produkt LA6-5DTNUB-RYGBC	17
2.3.2 Produkt XVC6B55S	17
2.3.3 Produkt 69916075	17
3 Teoretická část práce	18
3.1 Průmysl 4.0	18
3.1.1 Kyberneticko-fyzikální systémy	18
3.1.2 Základní principy	19
3.1.3 Další pojmy týkající se Průmyslu 4.0	20
3.2 Testbed samo-činný barman	20
3.3 Ethernet TCP/IP	22
3.4 Power over Ethernet	22
3.4.1 Topologie sítě s PoE	22
3.4.2 Přenos DC výkonu	23
3.4.3 Detekce a klasifikace koncového zařízení (PD)	23
3.5 Průmyslový komunikační protokol	26
3.5.1 Modbus	26
3.5.2 Modbus TCP	27
3.6 OMAC PackML	30
3.6.1 Stavový model podle ANSI/ISA-TR88.00.02-2015	30
3.6.2 Orientace ve stavovém diagramu	30
3.6.3 Význam definovaných stavů	32
3.6.4 Význam signalizovaných barev	34

4	Návrh Chytré signalizační jednotky	36
4.1	Koncepce zařízení	36
4.1.1	Zobrazovací módy	36
4.2	Koncepce HW	38
4.2.1	Popis funkce komponent v blokovém diagramu	38
5	Realizace Chytré signalizační jednotky	44
5.1	Návrh desky plošných spojů (DPS)	44
5.1.1	Proudové zatížení	47
5.1.2	Opravy desky plošných spojů	49
5.2	Bezpečnost zařízení	50
6	Návrh a implementace firmwaru SSU	52
6.1	Softwarové nástroje	52
6.1.1	ESP-IDF	53
6.1.2	Visual Studio Code	53
6.1.3	CMake	53
6.1.4	ModbusTool	53
6.2	Komunikace přes Ethernet	54
6.3	Modbus TCP server	56
6.3.1	Datový model	56
6.4	Řadič displeje	58
	Závěr	62
	Literatura	64
	Seznam symbolů a zkratek	67
A	Obsah elektronické přílohy	71

Seznam obrázků

2.1	Signální věž Schneider Electric řady Harmony XVC6	16
3.1	Testbed samo-činný barman – Sodovač	21
3.2	Testbed samo-činný barman – Zásobník sklenic	21
3.3	100BASE-TX Endspan napájení PD pomocí dvou párů	23
3.4	Průběh detekce, klasifikace a povolení plného výkonu do PD	25
3.5	Architektura Modbus TCP	27
3.6	Čtyři typy zpráv modelu klient/server	28
3.7	Zapouzdření dat žádosti/odezvy po Modbus TCP	28
3.8	Stavový diagram rozhraní podle standardu PackML	31
3.9	Typicky signalizované barvy pomocí majáků	34
4.1	Modulový zobrazovací mód	37
4.2	Font zobrazovaného textu na displeji	38
4.3	Blokový diagram Chytré signalizační jednotky	39
4.4	Modul ESP32-WROOM-32	40
4.5	Ethernet PHY – čip LAN8710A	40
4.6	Čip Si34061 pro správu napájení pomocí PoE	41
4.7	Osazený LED čip WS2812B	42
4.8	Detail LED čipu WS2812B	42
4.9	LED pásy WS2812B různých hustot osazení	42
5.1	Vrchní vrstva DPS – měď	45
5.2	Spodní vrstva DPS – měď	45
5.3	Příklad využití nástrojů Route a Differential Pair a Meander	46
5.4	Rozmístění komponent na vrchní vrstvě DPS	46
5.5	Rozmístění komponent na spodní vrstvě DPS s rozměry	46
5.6	3D model DPS – vrchní vrstva	47
5.7	3D model DPS – spodní vrstva	47
6.1	Blokový diagram firmwaru SSU	52
6.2	Příklad překladu ADU zprávy pomocí online parseru	54
6.3	Konfigurační registr z datového modelu SSU	56
6.4	Ukázka z implementované ASCII tabulky s fontem 5x7 pixelů	59
6.5	Zobrazování textu s vypnutými pixely pro text	60
6.6	Zobrazování textu se zapnutými pixely pro text	61

Seznam tabulek

2.1	Srovnání 3 vybraných signalizačních jednotek dostupných na trhu . . .	17
3.1	Porovnání standardů IEEE 802.3af a IEEE 802.3at	24
3.2	Výkonové třídy pro napájení dvěma páry vodičů	25
3.3	Výkonové třídy standardu IEEE 802.3bt	26
3.4	Vrstvový model Modbus TCP	28
3.5	Délka polí hlavičky MBAP	29
3.6	Modbus TCP – adresovací oblasti	29
4.1	Složení 24bitových dat pro LED čip WS2812B	43
6.1	Hodnoty konfiguračního registru	57
6.2	Hodnoty registru pro volbu názvu stavu	57
6.3	Hodnoty registru pro volbu animace	58
6.4	Význam bitů registru pro nastavení barvy	59

Úvod

Nikomu už asi není tajemstvím, že pokroky ve vědě za několik posledních desítek let jdou nesmírným tempem kupředu a zdá se, jako by neznaly hranic. Nároky technologických odvětví neustále rostou a nelze se tak vyhnout nutným reformám a inovacím. V průmyslu tomu pochopitelně není jinak a stále častěji tak lze narazit na již dnešní trend nazývaný Průmysl 4.0 – *Industry 4.0* (I 4.0). Ten zcela změnil dosavadní přístup k výrobě a hovoří se o něm jako o čtvrté průmyslové revoluci (viz 3.1).

Pro podpoření tohoto trendu a zároveň kvality výuky na fakultě elektrotechniky a komunikačních technologií VUT v Brně v laboratoři skupiny průmyslové automatizace tak vzniká *testbed*¹ nazývaný „Samo-činný barman“ (viz 3.2). Jeho činnost tak bude věrně znázorňovat typické vlastnosti a schopnosti systémů zaváděných v průmyslu v rámci již zmíněného Průmyslu 4.0.

Tato práce se zabývá kompletním návrhem koncepce i následnou realizací jak hardwarového, tak i programového řešení **Chytré signalizační jednotky**. Při návrhu je snahou držet se zásad a principů iniciativy Průmysl 4.0 i digitalizace.

Signalizační jednotka představuje jeden ze základních prvků strojů a autonomních jednotek k informování nebo varování příslušných osob o jejich okolí či stavu stroje, a **zastává** tak **funkci hlavního nositele vizuální**, ba dokonce zvukové, **zprávy pozorovateli**. Tomu tak dává najevo, jak by se měl zachovat nebo reagovat. Aby bylo dosaženo dostatečně rychlé a především správné reakce, musí být obsluha poučena, nebo alespoň řádně seznámena o významu jednotlivých signalizovaných stavů.

¹Jedná se o výrobní „poloprovoz“, který většinou slouží k vývoji a testování nových technologií za působení fyzikálních vlivů okolí.

1 Cíle práce

Tato bakalářská práce se podrobně zabývá celkovým návrhem konstrukce a programového řešení **Chytré signalizační jednotky**. Jak již bylo zmíněno v úvodu, jedná se o jeden ze základních prvků strojů pro zajištění informovanosti příslušných osob o jejich okolí či stavu stroje, a zajišťuje tak funkci hlavního nositele vizuální zprávy pozorovateli.

Jedná se tedy o signalizační **maják**, který zajišťuje jak běžné funkce, související se zobrazováním odpovídající barvy aktuálního stavu stroje podle standardu *Packaging Machine Language* (PackML), tak i několik dalších velmi užitečných funkcionalit. Mezi ty patří především komunikace prostřednictvím průmyslového komunikačního protokolu Modbus *Transmission Control Protocol* (TCP), napájení pomocí *Power over Ethernet* (PoE) nebo schopnost zobrazování v textového režimu s barevným pozadím (viz sekce 4.1.1).

Signalizační jednotka by měla představovat autonomní jednotku, která je schopna být zahrnuta do nadřazeného systému (sítě autonomních jednotek).

Dále je snahou držet se při návrhu a následné realizaci principů a zásad aktuální průmyslové revoluce.

1.1 Požadavky na Chytrou signalizační jednotku

Následující sekce jsou zaměřeny na definování požadavků na Chytrou signalizační jednotku. Ty jsou odvozené od současných trendů souvisejících s digitalizací a iniciativou Průmysl 4.0. Jak již bylo uvedeno v úvodu do této kapitoly, signalizační jednotka by měla být schopna zobrazovat stavy stroje podle standardu PackML, dále komunikovat přes Ethernet – konkrétně pomocí průmyslového protokolu Modbus TCP a být napájena pomocí standardu PoE.

1.1.1 Signalizace stavů

Tím nejpodstatnějším, co by měla signalizační jednotka umět, je pochopitelně signalizování stavů, ve kterých se aktuálně daný stroj/systém nachází. K nadefinování zobrazovaných stavů jsem využil standard PackML, protože vychází z dlouholeté praxe sdružení předních výrobců a je jakousi normou, které je žádoucí se držet. K těmto definovaným stavům jsem se navíc rozhodl přidat možnost nastavit si **vlastní požadovaný** (*custom*) **stav**, který je možné nastavit vhodným zápisem na adresy registrů komunikačního protokolu Modbus TCP. Každý stav bude signalizovaný jemu **příslušajícím nebo vlastně nadefinovaným textem o maximální délce 6 znaků**, který bude moct stát na místě nebo vizuálně téct zleva doprava či

zprava doleva po celém obvodu majáčku. Tento text bude zároveň podsvícen **stavu odpovídající nebo vlastně nadefinovanou barvou**.

1.1.2 Komunikace přes Modbus TCP

Z předchozího odstavce vyplývá, že by signalizační jednotka měla umět komunikovat prostřednictvím průmyslového komunikačního protokolu Modbus TCP (viz 3.5.2). Modbus TCP je velmi populární a již spoustu let často používaný protokol v oblasti průmyslové automatizace, především díky své jednoduchosti.

1.1.3 Napájení přes Power over Ethernet

Vzhledem k velikosti signalizační jednotky je mou snahou co nejvíce omezit počet přírodních kabelů. Toho chci dosáhnout využitím technologie PoE, která, jak popisují v podsekcí 3.4.1, zprostředkovává komunikaci i napájení koncového zařízení pomocí pouze jednoho Ethernetového kabelu kategorie (alespoň) CAT5.

1.1.4 Zajištění teplotní stability

Přestože se jedná o ještě poměrně malé zařízení, má při intenzivním svitu většiny pixelů poměrně velký odběr. Proto by měla obsahovat teplotní senzor a dostatečně výkonný ventilátor pro zajištění lepší teplotní stability. Jedná se tak o jeden z faktorů pro zajištění bezpečnosti zařízení (viz 5.2).

1.1.5 Konstrukce

Přibližné rozměry Chytré signalizační jednotky byly stanoveny na základě dohody s vedoucím práce: výška 100 mm a průměr 80 mm. Aby bylo možné těchto požadavků dosáhnout, rozhodli jsme se desku plošných spojů (DPS) navrhnout jako konstrukční prvek a řídicí elektronika se tak díky tomu zároveň nacházela v co nejtěsnější blízkosti.

Při volbě hardwarového vybavení signalizační jednotky je snaha držet se požadavků na autonomní zařízení.

2 Signalizační jednotka

Signalizační jednotka představuje základní prvek strojů či výrobních linek k informování nebo varování příslušných osob o jejich okolí či stavu stroje, a zastává tak funkci hlavního nositele vizuální zprávy pozorovateli, čímž signalizuje, jak by měl pozorovatel reagovat.

Signalizační jednotky mají ve většině případů válcovitý tvar a jsou upevněny ke stroji pomocí „stojanu“ nebo bočního držáku.

Kapitola popisuje základní parametry, mezi kterými je nutné volit při výběru požadované signalizační jednotky, a následně je proveden průzkum dostupných jednotek na trhu se srovnávací tabulkou.

2.1 Základní parametry

- **Průměr základny** (*diameter*) věže je běžně udáván v milimetrech. Pohybuje se od 25 mm až po 130 mm.
- **Výška** (*height*) věže závisí především na počtu modulů a zda je součástí i stojan/držák. Běžně je udávána v milimetrech a pohybuje se přibližně v rozmezí 120 mm až 330 mm.
- **Napětí** (*voltage*), kterým má být věž napájena. V průmyslu jsou typická napětí 24 V (AC/DC) nebo 100-230 V (AC/DC).
- **Počet barev** (*colors*) – signální věže bývají zpravidla modulární, takže si zákazník může vybrat, kolika-barevnou věž chce. Dodavatelé ale nejčastěji nabízejí konkrétní odstupňované modely produktů a není tak možné mezi sebou barvy libovolně prohazovat. Počet barev je nejčastěji v rozmezí 1 až 5: červená (R), oranžová (O), zelená (G), modrá (B) a čirá (C) – vždy v tomto konkrétním pořadí shora dolů. Dostupné jsou také signální věže např. s fialovým světlem.
- **Dostupné funkce** – některé modely umí blikající (*blink*) či zábleskový (*flash*) režim a některé pouze stálý (*steady*).
- **Bzučák** *buzzer* není nutnou součástí signálních věží. Intenzita hluku těchto majáček ve vzdálenosti 1 m se pohybuje okolo 70-85 dB. Hluk bývá buď kontinuální nebo přerušovaný.
- **Odběr proudu** – opět závisí na počtu modulů, přičemž na jeden svítící modul připadá až 300 mA (udáváno pro napájení 24 V).
- **Způsob montáže** – nejčastější jsou 3 způsoby: svislé připevnění na základnu (stojan), svislá přímá instalace (bez stojanu) a upevnění bočním držákem ke zdi.
- **Provozní teplota** – běžně se pohybuje v rozmezí nejméně -30°C až $+60^{\circ}\text{C}$.

- **Krytí** – nejčastější hodnoty krytí se pohybují v rozmezí IP54 (odolné proti prachu a stříkající vodě) až IP65 (pracho-těsné a odolné proti tryskající vodě).
- **PoE** – Signální věže napájené přes PoE a zároveň schopné komunikace po průmyslové sběrnici jsou velmi výjimečné (našel jsem jen pár kusů a jejich ceny se pohybovaly kolem 12ti tisíc Kč).

2.2 Průzkum trhu

Při prohledávání internetu, za účelem najít dostupné signalizační jednotky, jsem do prohlížeče zadával hesla jako: *signaling unit*, *stack lights*, *tower stack light*, *LED tower light*, *light towers* apod.

Krom běžných průmyslových majáků jsem našel pouze několik „bastlířů“, kteří se zabývali ve svých projektech konstrukcí vlastních signálních věží, nicméně konstrukce většinou spočívaly v pospojování několika desítek či stovek LEDek a vzniklý útvar ovládali pomocí několika integrovaných obvodů nebo rovnou využili některou, z dnes už velmi dostupných destiček ve stylu Arduino, Raspberry Pi ad. V žádném z těchto projektů jsem ale nenašel jakoukoli spojitost s využitím v průmyslu, napájením přes PoE či *Power over Ethernet Plus* (PoE+), komunikující pomocí průmyslového protokolu atd.

Za účelem provedení průzkumu trhu jsem se tedy omezil pouze na běžně dostupné světelné signální věže, které jsou velmi často používané v průmyslu např. na výrobních linkách. Takovým typickým designem může být signální věž řady Harmony XVC6 od Schneider Electric na obr. 2.1.



Obr. 2.1: Signální věž (XVC6) [1]

2.3 Srovnání vybraných produktů

Výrobci standardně dodávají signální věže s max. 5ti moduly (patry). Nejpřehlednější nabídku s možností srovnání jsem našel na <https://cz.rs-online.com>, odkud jsem vybral 3 parametrově odpovídající produkty a provedl jejich srovnání v tab. 2.1. Vybrané produkty jsou určeny k přímé montáži bez stojanu.

2.3.1 Produkt LA6-5DTNUB-RYGBC

K produktu z řady LA6 s označením LA6-5DTNUB-RYGBC se hlásí výrobce Patlite. Provedení zařízení působí velmi kvalitně a vzhledem k použitým RGB LEDkám se zdá být nejvíce variabilní v možnostech světelné signalizace.

2.3.2 Produkt XVC6B55S

Signální věž s označením XVC6B55SK spadá pod sortiment výrobce Schneider Electric. Provedení tohoto zařízení působí taktéž velmi kvalitně. Díky nižší základně dosahuje menší výšky. Věž podporuje jak stálý, tak i blikající svit. Jedním z mála negativ tohoto produktu je cena, která přesahuje 8 tisíc Kč.

2.3.3 Produkt 69916075

Produkt 69916075 od výrobce Werma je oproti vybraným produktům více jak o 2 cm užší. Podporuje pouze stálý svit, což může být v některých aplikacích omezující. Dosahuje nejvyššího odběru. Největším pozitivem je cena, která dosahuje pouhé poloviny ceny zmíněných produktů.

Tab. 2.1: Srovnání 3 vybraných signalizačních jednotek dostupných na trhu

Produkt:	LA6-5DTNUB-RYGBC	XVC6B55S	69916075
Provozní teplota	$-25\text{ }^{\circ}\text{C} - +60\text{ }^{\circ}\text{C}$	$-10\text{ }^{\circ}\text{C} - +50\text{ }^{\circ}\text{C}$	$-20\text{ }^{\circ}\text{C} - +50\text{ }^{\circ}\text{C}$
Průměr základny	60 mm	60 mm	37,5 mm
Výška	330 mm	280,8 mm	242,5
Světelný svit	různé efekty	blikající, stálý	stálý
Bzučák	•	•	•
Max. hlasitost	85 dB	85 dB	85 dB
Napájecí napětí	24 V DC	24 V AC/DC	24 V AC/DC
Stupeň krytí	IP54	IP54	IP65
Odběr proudu	210 mA	165 mA	300 mA
Barva pouzdra	stříbrná	bílá	černá
Přímá montáž	•	•	•
Cena za kus	7 408,64 Kč	8 162,00 Kč	3 977,55 Kč

3 Teoretická část práce

Následující podkapitoly a sekce se zabývají teoretickými oblastmi, které je nutné znát k porozumění této práce. První podkapitola se zabývá současným, velmi populárním trendem digitalizace nazývaným **Industry 4.0**, o kterém se mluví jako o 4. průmyslové revoluci. Dále je stručně popsán komunikační protokol **Ethernet TCP/IP** a podrobněji některé jeho standardy související se síťovou technologií **Power over Ethernet**. V poslední sekci se zabývám průmyslovým komunikačním protokolem **Modbus TCP**.

3.1 Průmysl 4.0

Tento název zastřešuje současný trend v oblasti automatizace a výměny dat v technologickém procesu výroby. Jde o celoevropskou iniciativu vládních činitelů a manažerů největších evropských firem, která má zapříčinit rapidní nárůst poptávky po nových – vyspělejších a modernějších technologiích. Zároveň tak chtějí urychlit vývoj robotizace a plně automatických řídicích systémů ideálně nezávislých na lidské obsluze.

Termín Industry 4.0 byl poprvé zmíněn na veletrhu v Hannoveru roku 2011 a jeho hlavní koncept a myšlenka byly představeny na stejném místě o 2 roky později [2]. Na jejich základě je také postavena myšlenka „**chytrých továren**“ (*smart factory*) jakožto propojení několika tzv. **kyberneticko-fyzikálních** (produkčních) systémů.

Při návrhu Chytré signalizační jednotky bylo snahou, aby zapadala do konceptu Průmyslu 4.0. V dalších sekcích je upozorněno na principy, které signalizační jednotka splňuje.

3.1.1 Kyberneticko-fyzikální systémy

V podstatě hlavní myšlenkou Průmyslu 4.0 je vytvoření inteligentní distribuované sítě (označované jako *kyberneticko-fyzikální systém* – *Cyber-Physical System* (CPS)) různorodých entit obklopující výrobní řetězec. Entity tvoří subsystémy, u kterých je snahou, aby byly schopny se autonomně (samostatně) rozhodovat, pracovat paralelně tzn. podle potřeby mezi sebou navzájem komunikovaly, a staly se tak samostatnými a plnohodnotnými členy komplexních výrobních celků. Každý fyzický systém by měl mít své **digitální dvojče** představující virtuální entitu, která zrcadlí život svého fyzického dvojčete, a slouží tak jako most mezi fyzickým a virtuálním světem. Aby bylo možné takové virtuální dvojče realizovat, musí být poskytnuto potřebné množství informací o fyzickém systému, čímž vznikne jeho digitální reprezentace.

3.1.2 Základní principy

Vědci identifikovali šest konstrukčních principů, které jsou pro podniky zásadní k efektivnímu využívání výhod Průmyslu 4.0.

Virtualizace je schopnost vytvářet virtuální modely na základě jednoho nebo několika fyzických systémech a vzájemně je propojovat. Díky tomu je následně možné simulovat scénáře „co kdyby...“, zobrazovat stav systému v reálném čase, analyzovat výkonnost a případně odhalit neefektivní operace nebo předcházet poruchám/haváriím.

Interoperabilita zajišťuje výměnu informací mezi komponenty, celými systémy i lidmi prostřednictvím IoT a IoS (podrobněji popsány dále). Obchodní informace lze následně sdílet mezi výrobními podniky, dodavateli a zákazníky.

Modularita a rekonfigurabilita je schopnost podniků flexibilně se přizpůsobovat měnícím se požadavkům a potřebám daného odvětví – tedy možnost nakoupit a implementovat pouze to, co je aktuálně nutné a v případě potřeby v budoucnu doplnit/rozšířit. Taková konfigurace umožňuje například mnohem snazší (rychlejší a efektivnější) přechod z výroby jedné rodiny výrobků na jinou.

Chytrá signalizační jednotka je schopna „za běhu“ měnit svoji konfiguraci, tedy počet modulů, zobrazujících určité stavy příslušnou barvou.

Decentralizace je snaha o rozdělení na jednotlivé subsystémy, které se rozhodují autonomně, paralelně a zároveň i nezávisle na ostatních systémech. Právě to hraje do karet již zmíněné modularitě.

Signalizační jednotka není k celkovému systému připojena pouze hardwarově, ale je schopna komunikace po síti, tzn. lze ji decentralizovat od zařízení, jehož stavy zobrazuje. Takové přemístění ale pochopitelně není v tomto případě žádoucí.

Schopnost reagovat v reálném čase zaručuje získání, nashromáždění a vyhodnocení dat, a následné vykonání akce v řádu milisekund, což umožňuje okamžitě reagovat na poruchy na výrobních linkách nebo proaktivně přistupovat k anomáliím nebo neefektivnosti.

Této vlastnosti je v případě Chytré signalizační jednotky dosaženo a umožňuje tak signalizovat např. výpadek komunikace nebo na jakém zařízení došlo k chybě.

Orientace na služby – díky propracovaným systémům a napojení na *big data* jsou podniky schopny se rychle adaptovat požadavkům zákazníků a poskytovat služby „na míru“.

Signalizační jednotka poskytuje stroji službu signalizování stavů a tedy splňuje tento princip.

Téměř veškeré informace v této kapitole jsem čerpal z [4].

3.1.3 Další pojmy týkající se Průmyslu 4.0

Již byly zmíněny pojmy chytrá továrna, kyberneticko-fyzikální systémy, digitální dvojčata a big data. Dále je uvedeno ještě několik pojmů, které se v oblasti Průmyslu 4.0 velmi často vyskytují. V Chytré signalizační jednotce nejsou následující pojmy uplatněny, nicméně v budoucích iteracích by neměl být veliký problém je implementovat.

„Internet věcí“ (*Internet of Things (IoT)*) – s přístupem na internet (resp. na cloud) mohou kyberneticko-fyzikální systémy spolupracovat a komunikovat mezi sebou a lidmi téměř v reálném čase. Z širšího hlediska se jedná o samostatnou velkou oblast využívající dnešní vyspělé techniky integrovatelné do malých embedded zařízení se schopností odesílat a přijímat data po síti bez nutné interakce s člověkem, přičemž každé zařízení je opatřené unikátním identifikátorem.

„Cloudové“ zpracovávání informací (*cloud computing*) – ke zpracovávání dat dochází na vzdáleném cloudu, kde je zprostředkováván potřebný úložný prostor, automaticky aktualizovaný software, kvalitní zabezpečení, vylepšená správa dokumentů a spousta dalšího bez nutnosti vlastnit tohle všechno na vlastní výpočetní technice. Zároveň je tak eliminována produkce uhlíkových stop organizací, jelikož využívají pouze takové množství zdrojů, jaké opravdu potřebují [5].

„Internet služeb“ (*Internet of Services (IoS)*) – je s ním úzce spojený „cloud computing“ a zastřešuje možnost poskytování aplikací, softwaru (dokonce i nástrojů k tvorbě aplikací a softwaru), úložiště a komunikace prostřednictvím cloudu.

Virtuální výroba je zjednodušeně řečeno výroba v počítači. Proces výroby probíhá podobně jako v reálném světě, přičemž výhodou virtuálních modelů je možnost simulace výroby a predikce potenciálních vad/poruch dříve, než se dostane do reálné výroby.

Digitální/smart výroba je simulace výrobních procesů, během které se využívají již získané znalosti a optimalizace procesů ještě před samotnou výrobou. Umožňuje tak zpětnou vazbu z aktuálních výrobních operací začleněním znalostí do procesu navrhování výrobku, resp. využije provozních znalostí během fáze plánování. Výsledkem je plnohodnotný funkční fyzický výrobek zhotovený v digitální továrně.

Chytrý produkt obsahuje potřebné informace týkající se výrobních kroků sebe samého a může být také schopen zaznamenávat důležité události svého životního cyklu.

3.2 Testbed samo-činný barman

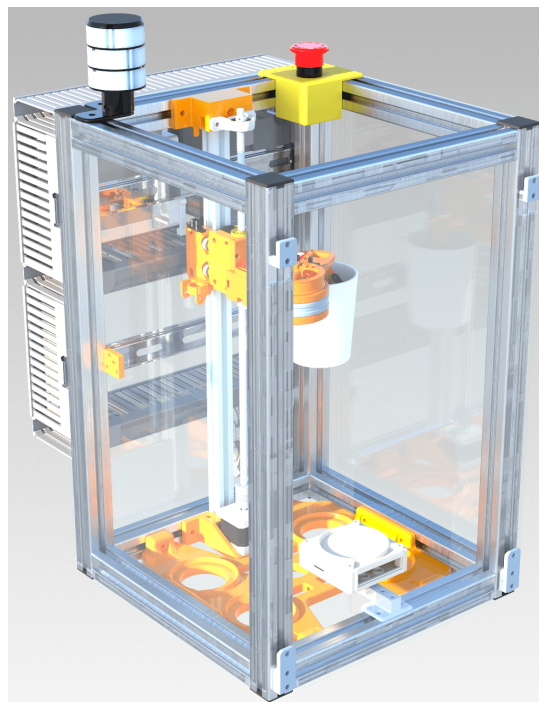
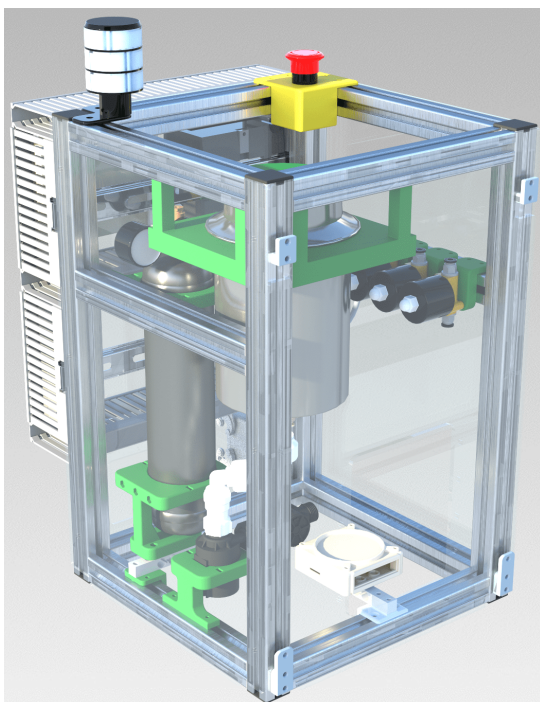
Testbedem se rozumí výrobní „poloprovoz“, který je obvykle umístěn ve výzkumné instituci, a na jejímž vývoji, financování i následném provozu se může podílet několik

firm. Většinou slouží k vývoji a testování nových technologií za působení fyzikálních vlivů okolí. Klíčovým aspektem testbedu je existence digitálního dvojčete.

Jedním z hlavních důvodů vzniku samo-činného barmana byla potřeba nové názorné studijní pomůcky pro výuku Průmyslu 4.0. Studenti se tak budou moci seznámit s trendy budoucí (resp. již aktuální) oblasti průmyslu.

Myšlenka automatizovaného barmana připravujícího nápoje vznikla vzhledem k atraktivitě a objektu zájmu studentů i širší veřejnosti. Proces míchání drinků je téměř každému dobře známý a je tak snazší a přirozenější pochopit testbedem demonstrované principy.

Na obrázcích 3.1 a 3.2 se nachází autonomní jednotky Sodovač a Zásobník sklenic, které jsou součástí barmana. **Lze si všimnout také nainstalovaných majáčků na obou jednotkách – ty budou nahrazeny Chytrou signalizační jednotkou, již se zabývám v této práci.**



Obr. 3.1: Testbed samo-činný barman – Sodovač Obr. 3.2: Testbed samo-činný barman – Zásobník sklenic

Podrobnější informace o samo-činném barmanovi můžete nalézt například v článku [6].

3.3 Ethernet TCP/IP

Ethernet TCP/IP je standardem lokálních sítí – *Local Area Network* (LAN) a zároveň z něj vycházejí varianty průmyslového Ethernetu. Jeho síťovou vrstvu (3. vrstva *Open Systems Interconnection* (OSI) modelu) tvoří protokol *Internet Protocol* (IP), který zajišťuje síťovou adresaci a směrování. Transportní vrstva (4. vrstva OSI modelu) je definována protokolem TCP a realizuje spojení mezi klienty a servery.

3.4 Power over Ethernet

Hlavní podstatou PoE je, jak už název napovídá, možnost zprostředkovávat po Ethernetu nejen komunikaci, ale zároveň i přenášet výkon potřebný pro napájení koncových zařízení. Tento fakt dokáže v průmyslu podstatně zkrátit dobu instalace, protože není nutné tahat další kabeláž potřebnou pro napájení. Dále není potřeba platit za kvalifikovaného pracovníka a oficiální kontrolu.

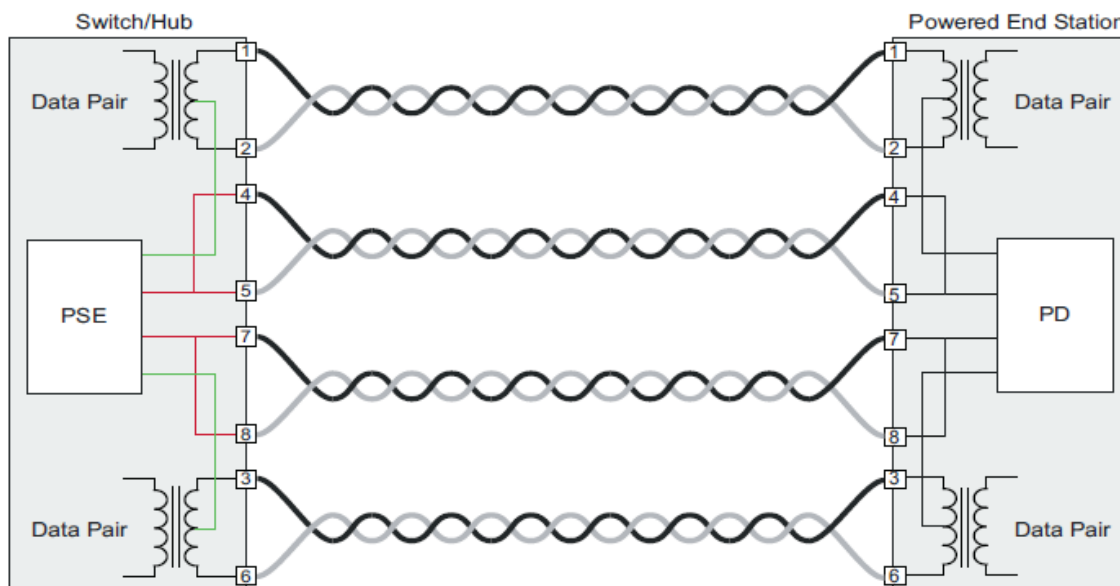
Jako první se dostal na trh standard **IEEE 802.3af-2003**, který byl schopen poskytnout *stejnoseměrný* – *Direct Current* (DC) výkon o pouhých **15,4 W** na jeden port. Tento fakt byl poměrně omezujícím kritériem a proto přišli vývojáři po šesti letech s dalším standardem **IEEE 802.3.at-2009** – **PoE+**, který umožňuje přenášet vyšší DC výkon pomocí dvou nevyužitých párů (*spare pairs*). Tento standard definuje 2 topologie.

3.4.1 Topologie sítě s PoE

V prvním případě (*endspan*) dodává napájecí DC výkon zařízení zprostředkovávající napájení (*Power-Sourcing Equipment* (PSE)) umístěné v přepínači/rozbočovači, zatímco druhá varianta (*midspan*) umožňuje tzv. *power insertion*, kdy je jednotka PSE mezi přepínačem/rozbočovačem a zařízením označovaným jako napájené koncové zařízení (*Powered Device* (PD)) [7].

Pro připojení PSE variantou power injection dále standard PoE+ definuje 2 alternativy. Alternativu A, kde je výkon dodáván přes středové odbočky oddělovacích transformátorů¹ na kroucené páry 1/2 a 3/6, a alternativu B, kde je výkon přenášen pomocí kroucených párů 4/5 a 7/8. Na obrázku 3.3 je zobrazeno připojení PSE topologií *endspan* vyhovující standardu 100BASE-TX (také známý jako IEEE 802.3u – Fast Ethernet).

¹Pro Ethernet je galvanické oddělení typické, a tento způsob patří k základním a účinným metodám pro zajištění funkční bezpečnosti.



Obr. 3.3: 100BASE-TX Endspan napájení PD pomocí dvou párů (Alternativa A – zelená; Alternativa B – červená) [7]

3.4.2 Přenos DC výkonu

V návaznosti na předchozí kapitolu a obr. 3.3 je vhodné zmínit způsob přenosu DC výkonu. Až do standardu 100BASE-TX, kdy probíhá komunikace pouze po dvou kroucených párech a zbylé 2 páry jsou nevyužité, se dal DC výkon posílat jednoduše po nevyužitém páru. Pro následující standardy (např. 1000BASE-T – Gigabit Ethernet) bylo potřeba vymyslet jiný způsob, jímž je právě Alternativa A (zelená) na obr. 3.3. Využívá se zde již zmíněných oddělovacích transformátorů se středovými odbočkami, kdy přenášený DC výkon je odebírán ze středové odbočky a komunikace je transformována z primárního vinutí na sekundární v poměru 1:1. Komunikace po Ethernetu navíc funguje na principu diferenciálních párů, takže „zvednutí“ napětí na obou vodičích každého páru na požadovanou úroveň 44-57 V nečiní žádný problém.

3.4.3 Detekce a klasifikace koncového zařízení (PD)

Aby bylo PSE schopno detekovat přítomnost PD, které podporuje napájení pomocí PoE, je nutné napříč každý napájecí pár připojit **25 k Ω detekční rezistor**. Díky tomuto zapojení nedojde k poškození zařízení nepodporující PoE, protože pokud PSE nedetekuje „správné“ zařízení, neaplikuje napájecí napětí. Dále tak PSE dokáže rozpoznat, jestli není PD odpojené od sítě [7].

PD během inicializace spojení rozpoznávají výkonovou třídu PSE (tzv. *classification*). Aby bylo možné úspěšně implementovat IEEE 802.3at zařízení, musí být

zajištěna kompatibilita se síťovými zařízeními IEEE 802.3af. Jinými slovy 802.3at PSE musí být schopné napájet 802.3af PD po kabelech CAT3 i CAT5, a stejně tak 802.3at PD musí umět detekovat, zda jsou napájeny z 802.3af nebo 802.3at PSE. Zařízení, která jsou kompatibilní pouze s 802.3af jsou klasifikovány jako **Typ 1** a prvky kompatibilní s 802.3at jako **Typ 2** [7]. Typ 1 je tedy parametrově ekvivalentní standardu IEEE 802.3af. Konkrétní elektrické parametry PSE a PD popisuje tab. 3.1.

Tab. 3.1: Porovnání standardů IEEE 802.3af „PoE“ a IEEE 802.3at „PoE+“ [9]

Parametr	PoE	PoE+ Type 2 - 2 páry
Výkon dostupný na PD	12,95 W	25,50 W
Max. výkon dodávaný PSE	15,40 W	30,00 W
Napěťový rozsah na PSE	44,0–57,0 VDC	50,0–57,0 VDC
Maximální proud I_{max}	350 mA	600 mA
Max. rezistivita kabelu na pár	20 Ω	12,5 Ω
Podporovaná kabeláž	CAT3 nebo lepší	CAT5 nebo lepší

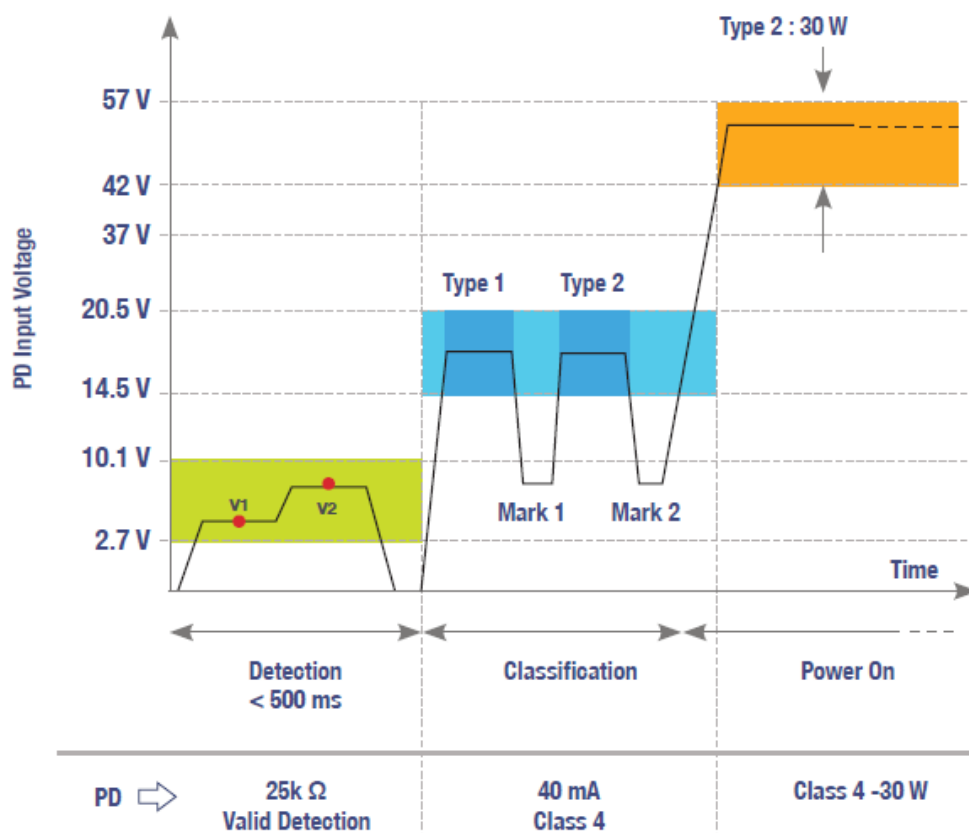
Zjistí-li PSE, že se jedná o PD (prezence detekčního rezistoru), musí ověřit výkonovou klasifikaci PD (potřebný napájecí výkon). Pro klasifikaci fyzické nebo linkové (spojové) vrstvy za účelem správy dodávaného výkonu do PD jsou dva mechanismy. Prvnímu mechanismu stačí znát elektrické vlastnosti PD a kabelu tvořícího spoj mezi PSE a PD. Druhý mechanismus využívá tzv. „protokol zjišťující koncový bod média“ (*Media Endpoint Discovery* (MED)), který je rozšířením Ethernetového „zjišťovacího protokolu na linkové vrstvě OSI modelu“ (*Link Layer Discovery Protocol* (LLDP)), tedy LLDP–MED. Dosahuje jemnějšího rozlišení výkonu (v krocích po 1,11 W) než první mechanismus a umožňuje PSE a PD podílet se na dynamickém přidělování výkonu, zatímco je PD v provozu. PD dále může periodicky sdělovat své energetické požadavky s PSE. Zařazení zařízení do jednotlivých tříd pro napájení pomocí dvou párů je uvedeno v tab. 3.2. Dále existují i pokročilejší techniky, které dosahují větší výkonové efektivity dynamickým snímáním aktuálního potřebného výkonu pro PD, případně spojením několika metod.

Průběh detekce, klasifikace a plynulého puštění plného výkonu do PD je ukázáno na obr. 3.4. V detekční fázi PSE aplikuje napětí 2,7 V až 10,1 V a měří odběr proudu připojeného PD. Je-li v PD přítomen detekční rezistor mezi piny VPOS a RDET, fáze proběhne v pořádku a pokračuje klasifikační fází. PSE tentokrát aplikuje napětí 15,5 V až 20,5 V a opět měří odebíraný proud připojeného PD. Zjištěná hodnota proudu, která je dána klasifikačním odporem připojeným mezi piny RCL a VNEG, určí třídu PD. Pokud PSE detekuje proud v rozsahu odpovídající třídám 0-3, apli-

Tab. 3.2: Výkonové třídy pro napájení dvěma páry vodičů [7]

Výkonová třída	Maximální průměrný výkon dostupný na PD
Třída 0	13,00 W
Třída 1	3,84 W
Třída 2	6,49 W
Třída 3	13,00 W
Třída 4	25,50 W

kuje plný výkon pro *Type 1* PD. Pakliže proud odpovídá rozsahu pro třídu 4 (36 až 44 mA), je zahájena klasifikace pro *Type 2* PD, což znamená, že PSE sníží napětí na úroveň označenou *Mark 1* (7 V až 10 V), a znovu opakuje sekvenci klasifikace a „značkování“ (*Mark 2*). Tento proces je nazýván „dvou-událostní klasifikace“. Proběhne-li v pořádku, je povolna aplikován plný výkon odpovídající *Type 2* PD. PSE následně průběžně kontroluje přítomnost PD snímáním odebíraného proudu, který musí být alespoň 10 mA v časovém intervalu 300 ms. Splnění této podmínky se dá dosáhnout i využitím tzv. *Maintain-Power-Signature* (MPS), což je vlastnost PD, která záměrně navyšuje odběr generováním proudových pulzů.



Obr. 3.4: Průběh detekce, klasifikace a povolení plného výkonu do PD [8]

Na konci září roku 2018 byl schválen standard **IEEE 802.3bt**, který už dokáže po všech čtyřech kroucených párech přenášet až obdivuhodných **90 W**. Tento standard už není v rámci této práce popsán, ale pro představu je zde uvedena tab. 3.3, která ilustruje, kam až se za 15 let tato technologie posunula.

Tab. 3.3: Výkonové třídy standardu IEEE 802.3bt

PSE	Typ 3 (802.3bt)						Typ 4 (802.3bt)	
	Typ 1 (802.3af)			Typ 2 (802.3at)				
	Třída 1	Třída 2	Třída 3	Třída 4	Třída 5	Třída 6	Třída 7	Třída 8
	4 W	7 W	15,4 W	30 W	45 W	60 W	75 W	90 W
PD	Přenos pouze dvěma páry (Typ 1 & 2)				Přenos vždy všemi čtyřmi páry			
	Přenos dvěma nebo čtyřmi páry (Typ 3 & 4)							
	Třída 1	Třída 2	Třída 3	Třída 4	Třída 5	Třída 6	Třída 7	Třída 8
	3,84 W	6,49 W	13 W	25,5 W	40 W	51 W	62 W	71,3 W

Při realizaci Chytré signalizační jednotky bude použit standard IEEE 802.3.at-2009 (PoE+, PoE Plus) typu 2 třídy 4 v topologii *endspan*, který svými parametry postačí a zároveň nebude realizaci zbytečně prodražovat ještě lepší technologií.

3.5 Průmyslový komunikační protokol

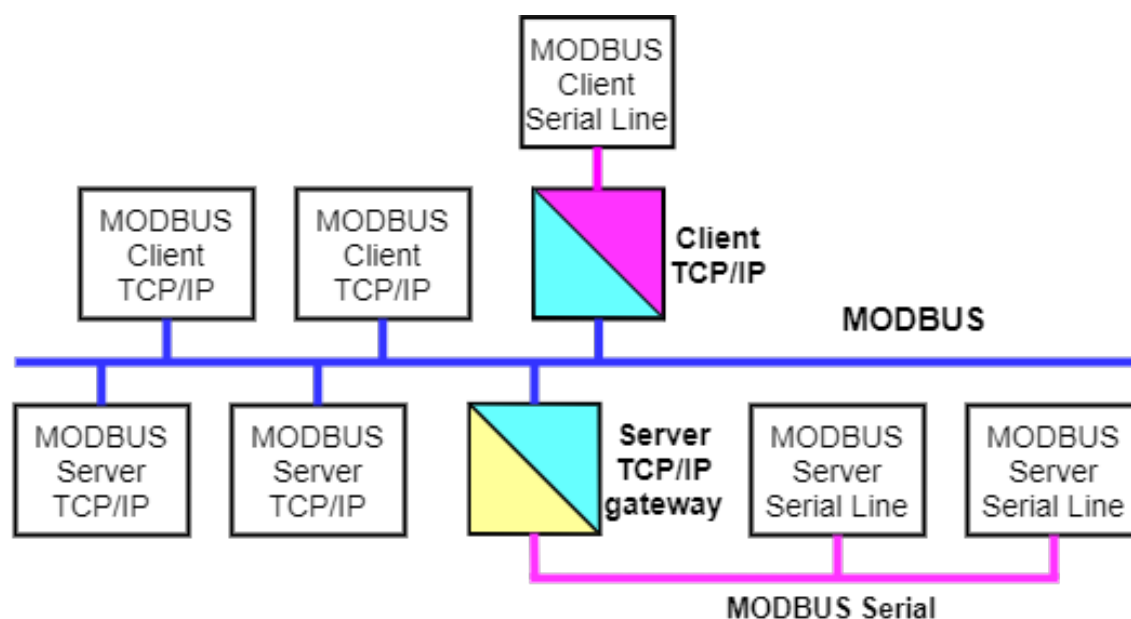
Na úvod této kapitoly je nutné zmínit, že průmyslový protokol Modbus TCP, který je použit v návrhu Chytré signalizační jednotky, zcela neodpovídá konceptu Průmyslu 4.0, a to např. z toho důvodu, že to není architektura orientovaná na služby a neumožňuje horizontální integraci. Jelikož se jedná o první iteraci návrhu, byl použit Modbus TCP především kvůli jeho jednoduchosti. Vhodnějšími protokoly by mohly být např. velmi známý *Open Platform Communications Unified Architecture* (OPC UA) nebo *Message Queuing Telemetry Transport* (MQTT) jakožto představitel komunikace „vydavatel/odběratel“ (*Publisher/Subscriber*).

3.5.1 Modbus

Modbus je průmyslový komunikační protokol publikovaný v roce 1979 společností Modicon (dnes Schneider Electric), který byl původně vytvořen se záměrem zprostředkovat datovou komunikaci mezi jejich programovatelnými automaty (*Programmable Logic Controller* (PLC)). V průmyslu je populární díky své volné dostupnosti a jednoduchosti. Často je využíván k propojení dohlížejícího (*supervisory*) počítače se vzdálenou koncovou jednotkou (*Remote Terminal Unit* (RTU)) uvnitř systémů pro správu, dohled a řízení (*Supervisory Control and Data Acquisition* (SCADA)). Je považován za „dědečka“ moderních průmyslových sběrnic (*fieldbuses*) [10].

3.5.2 Modbus TCP

Modbus (TCP) je typicky implementován na sítích Ethernet, k níž jsou připojeni klienti (*Modbus clients*) a servery (*Modbus servers*) prostřednictvím své IP adresy. Z toho už může být patrné, že funkčním základem tohoto protokolu je obecně známý Ethernet TCP/IP (viz 3.3). Modbus TCP byl vyvinut s nástupem průmyslového Ethernetu, přičemž v obálce TCP/IP zachovává většinu jednoduchosti Modbusu [11]. Samozřejmostí je připojení několika zařízení k jedné síti (viz obr. 3.5).



Obr. 3.5: Architektura Modbus TCP, převzato a upraveno z [12]

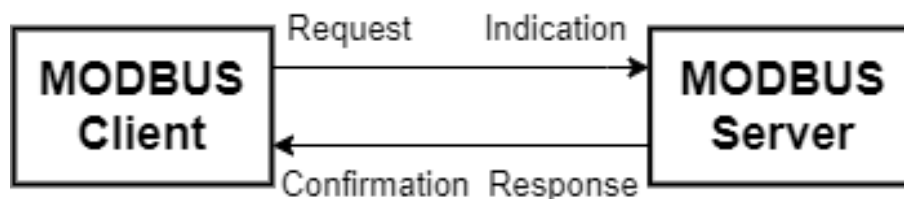
Popis funkce standardu Modbus TCP

V porovnání s referenčním 7-vrstvým *International Organization for Standardization* (ISO)/OSI modelem Ethernetu je Modbus TCP definován pouhými čtyřmi vrstvami. První vrstva (*Network access* (Ethernet)) zastupuje funkci 1. a 2. vrstvy OSI modelu. Druhá vrstva (*IP*) je ekvivalentní vrstvě (*Network*) a třetí odpovídá vrstvě (*Transport*) OSI modelu. Poslední vrstvu (aplikační) definuje protokol Modbus, který je nezávislý na přenosovém médiu a zastupuje zbylé vrstvy OSI modelu. Pro lepší představu lze nahlédnout do tab. 3.4.

K datovým přenosům dochází mezi klienty a servery. Protokol definuje 4 typy zpráv: **žádost** (*request*), **potvrzení** (*confirmation*), **indikace** (*indication*) a **odezva** (*response*) - jak je znázorněno na obr. 3.6. Přesně v tomto pořadí také probíhá komunikace.

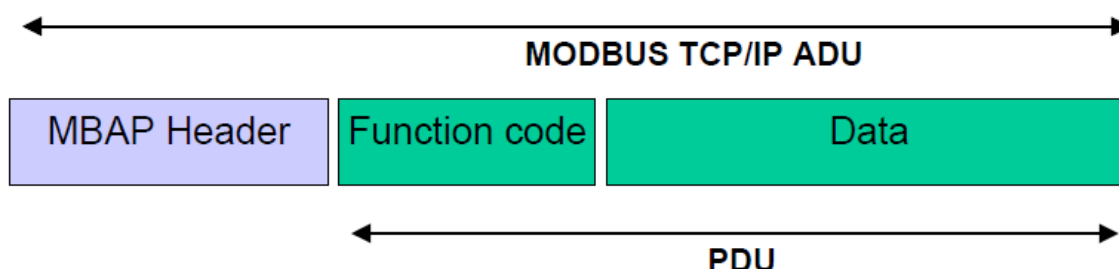
Tab. 3.4: Vrstvový model Modbus TCP, převzato a upraveno z [13]

Vrstva	ISO/OSI funkce	Modbus TCP funkce
5, 6, 7	aplikační	Aplikační protokol Modbus
4	transportní	Transmission Control Protocol (TCP)
3	síťová	Internet Protocol (IP)
2	spojová	IEEE 802.3: Ethernet
1	fyzická	



Obr. 3.6: Čtyři typy zpráv modelu klient/server, převzato a upraveno z [12]

Protokol Modbus dále definuje datovou jednotku (*Protocol Data Unit (PDU)*) a při mapování protokolu na konkrétní sběrnice nebo sítě může zavést některá další pole na aplikační datové jednotce (*Application Data Unit (ADU)*) (viz obr. 3.7).



Obr. 3.7: Zapouzdření dat žádosti/odezvy po Modbus TCP, převzato z [12]

Data jsou založena na 16-bitových registrech, které mohou obsahovat diskrétní hodnotu 1 (*on*)/ 0 (*off*) nebo 16-bitovou celočíselnou hodnotu. Některé implementace používají 2 nebo více celočíselných registrů pro reprezentaci hodnot s plovoucí desetinnou čárkou nebo dlouhých celočíselných hodnot [11].

Klient zahajující transakci vytvoří ADU, kde funkční kód (*function code*) určuje serveru, jakou akci má provést. Klienti si mohou od serverů vyžádat diagnostická data a servery mohou klientům posílat chybové kódy (*error codes*), když zjistí, že něco není v pořádku s žádostí, kterou obdržely [11].

Tab. 3.5 popisuje délku jednotlivých polí uvnitř **hlavičky MBAP** (MODBUS Application Protocol).

Tab. 3.5: Délka polí hlavičky MBAP, převzato a upraveno z [12]

	Typ transakce	Protokol	Délka	Identifikátor jednotky
Velikost	2 B	2 B	2 B	1 B

Hlavička MBAP je „dlouhá“ 7 B. **Typ** (identifikátor) **transakce** je používán pro párování transakcí – Modbus server zkopíruje do odpovědi identifikátor transakce požadavku. **Identifikátor protokolu** (Protokol) se používá pro multiplexování uvnitř systému. **Protokol Modbus** je identifikován hodnotou **0**. **Délka** udává počet bajtů následujících polí, včetně identifikátoru jednotky a datových polí. **Identifikátor jednotky** slouží pro účely směrování uvnitř systému. Typicky se používá ke komunikaci s Modbus+ (založeném na topologii sítě *token ring*) nebo podřízenou sériovou linkou Modbus (přes bránu mezi sítí Ethernet TCP/IP a sériovou linkou Modbus). Toto pole je nastaveno klientem v požadavku a musí být vráceno se stejnou hodnotou v odpovědi serveru. [12]

Modbus TCP je v zásadě Modbus RTU zabalený v balíčku Ethernet (IEEE 802.3) s cílovou (IP) adresou pomocí transakčního protokolu TCP/IP. Pro Modbus je vyhrazen **TCP port 502**, zatímco nový Modbus/TCP Security používá port 802 [11].

Standardní datový model

Data jsou uložena v zařízení Modbus server, jehož adresový prostor (datový model) je standardně rozdělen do čtyř různých oblastí pomocí příslušejících odkazů (referencí): 0xxxx, 1xxxx, 3xxxx a 4xxxx. Každé oblasti náleží jeden ze čtyř základních datových typů: **diskrétní vstupy** (*discrete inputs*), **cívky** (*coils*) – neboli diskrétní výstupy, **vstupní registry** (*input registers*) a **uchovávací registry** (*holding registers*). V tab. 3.6 jsou k nim doplněny také informace o jejich velikosti nebo možnosti přístupu [14].

Tab. 3.6: Modbus TCP – adresovací oblasti, převzato a upraveno z [14]

Oblast	Velikost	Přístup	Poznámka
Cívky (<i>Coils</i>)	1 bit	čtení/zápis	lze změnit aplikačním programem
Diskrétní vstupy		pouze čtení	poskytnuto I/O systémem
Vstupní registry	16-bitové	pouze čtení	poskytnuto I/O systémem
Uchovávací registry	slovo	čtení/zápis	lze změnit aplikačním programem

Pro oblasti je také specifický jejich **posun** (*offset*) – 1, 10001, 30001 a 40001 – ve stejném pořadí, jak jsou uvedeny v tab. 3.6.

Oblast vždy obsahuje 9999 hodnot a všechny hodnoty v rámci jedné oblasti tak mají svou **datovou adresu** v rozsahu 0x0000 – 0x270E (0 – 9998). Právě tyto adresy jsou pak obsaženy v Modbus zprávách. To znamená, že například uchovávací registr s číslem 40005 má svou datovou adresu 0x0004.

3.6 OMAC PackML

V této podkapitole je podrobněji popsán obecný stavový diagram automatizované jednotky, vycházející ze standardu PackML, a vysvětlen význam jednotlivých stavů.

Organizace *Organization for Machine Automation and Control* (OMAC) byla založena za účelem pomoci výrobcům zapojit se do společné práce a hledání inovací, a dosáhnout tak efektivnější produkce, kratších doručovacích dob, zvýšení ziskovosti ad. OMAC spojila přední výrobce a umožnila jim tak sdílet mezi sebou nejlepší praktiky, aby z toho mohlo těžit celé odvětví průmyslu, a to vše při zachování jejich konkurenční výhody. OMAC je zároveň společně s *International Society of Automation* (ISA) zakládajícím členem Automation Federation (známá také jako „*Voice of Automation*“) – „sdružení členských organizací poskytujících povědomí, programy a služby, které neustále rozvíjejí profesi automatizace pro zlepšení lidstva“².

3.6.1 Stavový model podle ANSI/ISA-TR88.00.02-2015

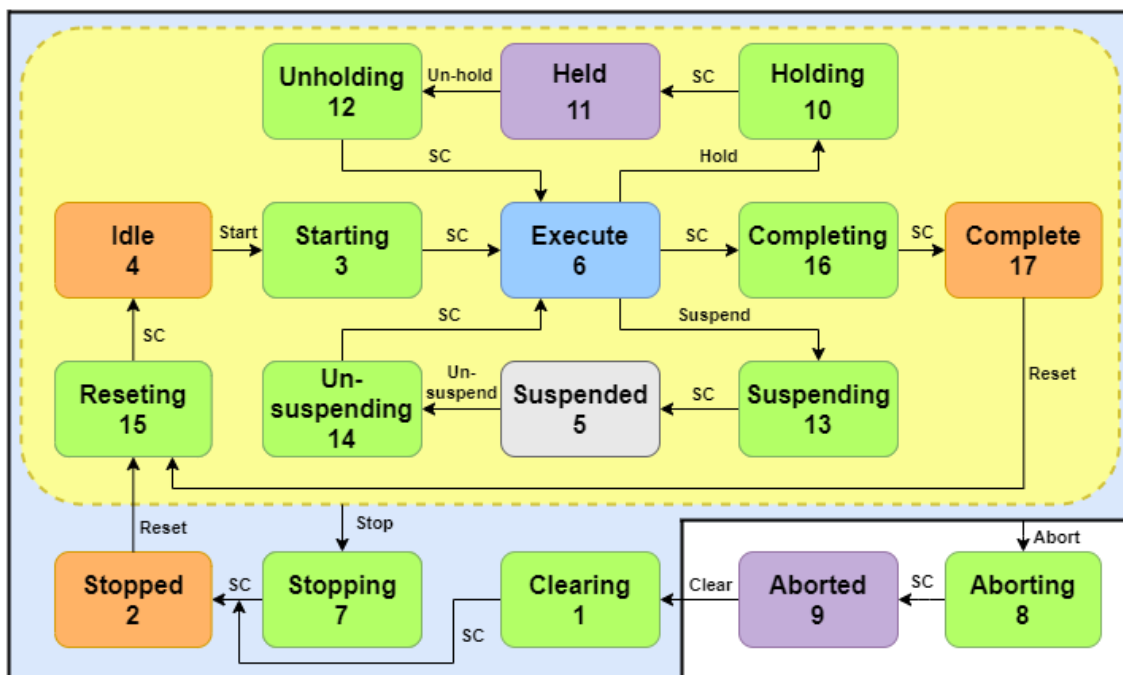
PackML unit/machine Implementation Guide je první část technického reportu ANSI/ISA-TR88.00.02-2015 organizace OMAC obsahující nejlepší doporučené praktiky, které mají pomoci společností s rozdílnými úrovněmi znalostí implementovat rozhraní na automatizovaných jednotkách či strojích. Report definuje jednotnou terminologii proměnných použitých v řídicích programech a manažera stavů rozhraní, který obsahuje všechny potřebné stavy (viz obr. 3.8). Dále poskytuje několik módů, ve kterých může stroj fungovat – manuální, produkce, údržba a uživatelské (testovací) módy. Pro každý mód je nadefinovaný diagram stavů a přechodů mezi nimi.

V celé této sekci jsem čerpal z [15].

3.6.2 Orientace ve stavovém diagramu

Před samotným popisem jednotlivých stavů je pro lepší pochopení ve zkratce vysvětlen význam bloků a popisků orientovaných šipek ve stavovém diagramu (viz obr. 3.8).

²Dostupné z: <<https://www.automationfederation.org/About/About>>



V následujících odstavcích používám pojmy **jednotka** a **systém**. Pod jednotkou je možné si představit například podavač materiálu uvnitř celého systému, jímž je například výrobní linka.

Zelené bloky představují **akční stav**, tzn. provádějí nějakou akci v rámci jednotky. Do těchto bloků vždy vstupuje **příkaz**, který spouští daný akční stav. Po **dokončení akce** (*State is Completed* (SC)) dojde k přechodu z akčního stavu buď do **čekajícího stavu** (bloky vyplněny oranžovou/fialovou/šedou barvou), nebo opět do akčního stavu *Execute*.

Jediný modře vyplněný stav Execute reprezentuje stabilní akční stav jednotky, v němž provádí svoji hlavní činnost.

Oranžově vyplněné jsou stabilní čekající stavy k identifikaci, že jednotka dosáhla jistého souboru podmínek. Jednotka v těchto stavech setrvává dokud nedostane příkaz k přechodu do některého z akčních stavů.

Zeleně vyplněné stavy představují určitou akci (činnost) jednotky, například zvyšování rychlosti. Jde o jednorázové nebo opakované provedení kroků v logickém pořadí, v daném časovém úseku nebo do dosažení určité podmínky. Například v počátečním stavu se zkontroluje kvalita a platnost získaných dat, než dojde ke zvýšení rychlosti vykonávání dané činnosti.

Fialově vybarvené stavy zastupují situaci, kdy se jednotka nachází v poruchovém režimu generujícím alarm nebo varování. Jednotka nevykonává svoji hlavní činnost dokud operátor nezajistí přechod do stavu Execute, například odstraněním

zaseknutého (případně doplnění potřebného) materiálu nebo bezpečným napravením zařízení, které poruchu způsobilo.

Šedě vyplněný stav představuje případ, při kterém jednotka nevykonává hlavní činnost. Buď přestane vykonávat činnost úplně, nebo pokračuje v cyklu bez vykonávání hlavní činnosti, dokud se podmínky externího procesu nevrátí do normálu, obvykle bez jakéhokoli zásahu operátora.

Stavy ohraničené čárkovanou čarou (oblast vyplněná žlutě) mohou být přerušeny příkazem *Stop*, čímž dojde k přechodu do akčního stavu *Stopping* a následně do čekajícího stavu *Stopped*. Z tohoto stavu se dá dostat pouze příkazem *Reset*, po němž následuje akční stav *Resetting* a potom čekající stav *Idle*.

Stavy ohraničené plnou čarou (oblast vyplněná modře) mohou být přerušeny příkazem *Abort*, čímž dojde k přechodu do akčního stavu *Aborting* a následně do čekajícího stavu *Aborted*. Odtud se lze dostat pouze příkazem *Clear*, po němž následuje akční stav *Clearing* a potom opět čekající stav *Stopped*. Postup ze stavu *Stopped* do stavu *Idle* je stejný jako na konci předešlého odstavce.

3.6.3 Význam definovaných stavů

- ***Idle*** – dává najevo, že resetování bylo dokončeno. Jednotka je tak připravena na obdržení nastavovacích parametrů. Pokud nastane porucha, přejde přes stav *Stopping* do stavu *Stopped*. Ve stavu *Idle* jednotka zaručuje bezpečné chování (nečinnost) a připravenost k vykonávání hlavní činnosti.
- ***Starting*** – přechod do tohoto stavu je zahájen příkazem *Start*. Zajišťuje nutné kroky pro spuštění jednotky. Jednotka načte počáteční parametry a pakliže byly načteny správně, přejde do stavu *Execute*. Při načtení chybných parametrů jednotka přejde přes stav *Stopping* do stavu *Stopped*. Jednotka nepřejde do stavu *Execute*, dokud neproběhne bezchybně předchozí postup. Na konkrétní funkci tohoto stavu se musí dohodnout dodavatel a koncový zákazník.
- ***Execute*** – jednotka vykonává svoji hlavní činnost a prochází všechny definované kroky podle zvoleného módu a receptu.
- ***Complete*** – tomuto stavu předchází stav ***Completing***, během kterého se provádí nutné kroky k dokončení činnosti a zároveň jsou zaznamenány potřebné hodnoty (např. počet vyrobených produktů). Pokud operátor změnil během procesu *Execute* některé parametry, stávají se v průběhu tohoto procesu dostupné zbytku systému. Konkrétní funkce závisí na dohodě dodavatele s koncovým zákazníkem. Po dokončení stavu *Completing* přechází jednotka do stavu *Complete*. Ten signalizuje dokončení činnosti a čekání na příkaz *Reset*.
- ***Resetting*** – obdržetím příkazu *Reset* přejde jednotka do stavu *Resetting*, čímž typicky vymaže (vyčistí) data (recept a parametry) a nastaví jednotku

do stavu Idle, ve kterém čeká na příkaz Start. Na konkrétní funkci tohoto stavu se opět musí dohodnout dodavatel a koncový zákazník.

- **Held** – hlavní činnost jednotky je **přerušena z důvodu interní podmínky**, která nijak nesouvisí s okolními jednotkami uvnitř systému. Může jít například o doplnění materiálu potřebného k činnosti – je tedy potřeba alespoň drobného zásahu operátora. Aktuální nastavení a stavy jednotky v okamžik obdržení příkazu *Hold* je nutné uložit během řízeného zastavení **Holding**, pro následný přechod (příkazem Un-hold) přes stav **Unholding** do stavu vykonávání hlavní činnosti (Execute), kdy není žádoucí jít přes stavy Aborted a Stopped.
- **Suspended** – činnost jednotky je **přerušena z důvodu externí podmínky** bránící v pokračování. Ty většinou vznikají uvnitř systému, v němž se jednotka nachází, nebo je vyvolá operátor přes *Human-Machine Interface* (HMI). Stejně jako v předchozím případě, jedná se o řízené zastavení během stavu **Suspending**, v němž je nutné uložit nastavení a stavy jednotky v okamžiku obdržení příkazu *Suspend*. Na konkrétních speciálních podmínkách, při kterých dojde k přechodu do stavu Suspending, se musí dohodnout dodavatel a koncový zákazník. Ze stavu Suspended, pro návrat k vykonávání hlavní činnosti (Execute), dojde příkazem Un-suspend přes stav **Un-suspending**.
- **Stopped** – příkazem Stop je způsoben přechod do stavu **Stopping** zajišťující řízený přechod do stavu Stopped. Na konkrétní funkci tohoto stavu se musí dohodnout dodavatel a koncový zákazník. Například je-li nutné zastavit ihned nebo je možné ještě dokončit některé procesy. Ve stavu Stopped je jednotka úplně pozastavena, nicméně dále může normálně komunikovat se zbytkem systému, ale už není možné spustit proces příkazem Start. Jednotka čeká na příkaz Reset, což je jediný možný způsob přechodu do jiného stavu.
- **Aborted** – tomuto stavu předchází stav **Aborting**, do nějž je možné se dostat **kdykoli** příkazem Abort a představuje okamžité bezpečné zastavení jednotky. Na konkrétních nutných akcích, při kterých dojde k přechodu do stavu Aborted, se musí dohodnout dodavatel a koncový zákazník. Jednotka by si v tomto stavu měla alespoň udržet informaci o jejím aktuálním stavu. Opustit ho může pouze po resetování detekovaného bezpečnostního obvodu (například uvolnění Emergency-stop (E-stop) tlačítka) a následném příkazu Clear.
- **Clearing** – Operátor „vyčistí“ bezpečnostní systém a následně uvede jednotku do stavu Stopped. Jednotka vymaže poruchy, k nimž mohlo dojít při přerušení, a jsou zobrazeny ve stavu Aborted, než přejde do stavu Stopped. Jednotka zpřístupní své parametry zbytku systému. Parametry stroje jsou k dispozici za účelem získání informací o případných manuálních změnách a opravách dat na jednotce. Konkrétní funkce závisí na dohodě dodavatele s koncovým zákazníkem.

3.6.4 Význam signalizovaných barev

Signalizování barev příslušejících konkrétním stavům je hlavní funkcí signalizační jednotky. Určení nejvhodnější barvy závisí na zprávě, kterou barva vyjádří pozorovateli na základě geografických nebo kulturních norem. Mezi nejběžnější barvy používané pro majáky patří červená, oranžová/jantarová/žlutá, modrá, zelená, bílá a fialová (viz obr. 3.9³). Význam barev se může v různých aplikacích mírně lišit, nicméně alespoň červená, jantarová a zelená barva by měly zůstat zachovány, aby nedošlo k nežádoucímu zmatení obsluhy.



Obr. 3.9: Typické signalizované barvy pomocí majáků

Následující seznam významů barev je převzat od firmy EATON⁴. Na obr. 4.1 v sekci 4.1.1 je znázorněn způsob zobrazování jednotlivých barev na *Smart Signaling Unit* (SSU) při modulovém módu.

- **Červená – NEBEZPEČÍ** – bývá používána jako varování před nebezpečím. To znamená, že obsluha by měla zastavit nebo uhnout z cesty do bezpečí. V případě že světlo bliká, červená vysílá velmi silný neverbální signál o bezprostředním nebezpečí.
- **Jantarová (žlutá) – BUĎTE SI VĚDOMI** – se používá jako varování, že hrozí nebezpečí. Operátor by měl být ve střehu a vědom svého okolí. Někdy se používá k upozornění operátora na potenciální nebezpečí jímž může být například nízká viditelnost nebo že se zařízení chystá aktivovat (automatické dveře).
- **Modrá – ZMĚNA** – většinou naznačuje, že se pozorovatel musí dostat z cesty a někdy se používá jako alternativa k červené. Ve výrobě modrá často signali-

³Převzato z:

<<https://www.eaton.com/content/dam/eaton/products/safety-security-emergency-communications/fire/fulleon/x10/images/x10-full-beacon-range-1400-900.png.transform/default-desktop/image.jpg>>

⁴Převzato a upraveno z:

<<https://www.eaton.com/gb/en-gb/products/safety-security-emergency-communications/alarms-and-signaling-devices/beacon-colours-for-industrial-signalling.html>>

zuje rozdílnost, jako je podmíněná reakce na napájení procesů kritických pro bezpečnost.

- **Zelená – PROBÍHÁ AKCE** – obvykle znamená, že vše běží dobře a neexistují žádné problémy s procesem nebo podmínkami prostředí.
- **Bílá – VELKÁ VZDÁLENOST** – představuje nejjasnější světlo, takže se používá, když je potřeba vysílat indikaci na delší vzdálenosti a někdy tak nahrazuje červený maják. Při špatné viditelnosti by mohla být použita jako indikátor pro zvýraznění různých oblastí nebo zařízení v případě velký pohyblivých objektů.
- **Purpurová (fialová) – NEBEZPEČÍ** – může být někdy použita jako alternativa k červené pro označení konkrétního nebezpečí, jako je například požár. Navrhuje, aby si lidé byli vědomi bezprostředního nebezpečí, a aby podnikli kroky, jako je opuštění oblasti. Často se používá v ropném a plynárenském průmyslu k signalizaci extrémně vysokých úrovní hořlavého plynu.

4 Návrh Chytré signalizační jednotky

Nadefinováním požadavků, které by měla Chytrá signalizační jednotka (dále jen SSU) zajišťovat, jsem se zabýval již v podkap. 1.1. V následujících podkapitolách je popsán, a pomocí blokového diagramu předveden, základní koncept SSU. Následně jsou vysvětleny funkce jednotlivých komponent v blokovém diagramu.

4.1 Koncepce zařízení

Vzhledem k funkcím SSU se jedná z obecného pohledu o displej, který, jak již bylo zmíněno v kap. 2, představuje objekt válcovitého tvaru. Tento válec je tvořen sedmi, nad sebe rozloženými, digitálními *Light-Emitting Diode* (LED) pásy (viz sekce 4.2.1), na nichž je zobrazována barva příslušející požadovanému stavu, a v případě textového módu (viz sekce 4.1.1) i odpovídající text.

4.1.1 Zobrazovací módy

Signalizační jednotka podporuje 2 zobrazovací módy – **modulový** a **textový**. Jejich princip je vysvětlen v následujících podsekcích. Volba požadovaného módu a konfigurace se provádí zápisem do odpovídajícího holding registru (`CConfiguration_t`) Modbus serveru uvedeného v sekci 6.3.1, kde lze nalézt význam a strukturu všech ostatních datových typů definujících celkový datový model.

Oba módy (i jejich režimy) mají společné to, že si volí z předdefinované množiny názvů stavů (podle *Identifier* (ID)) na základě standardu PackML (viz 3.6). K tomu je navíc možné dodefinovat si několik svých stavů.

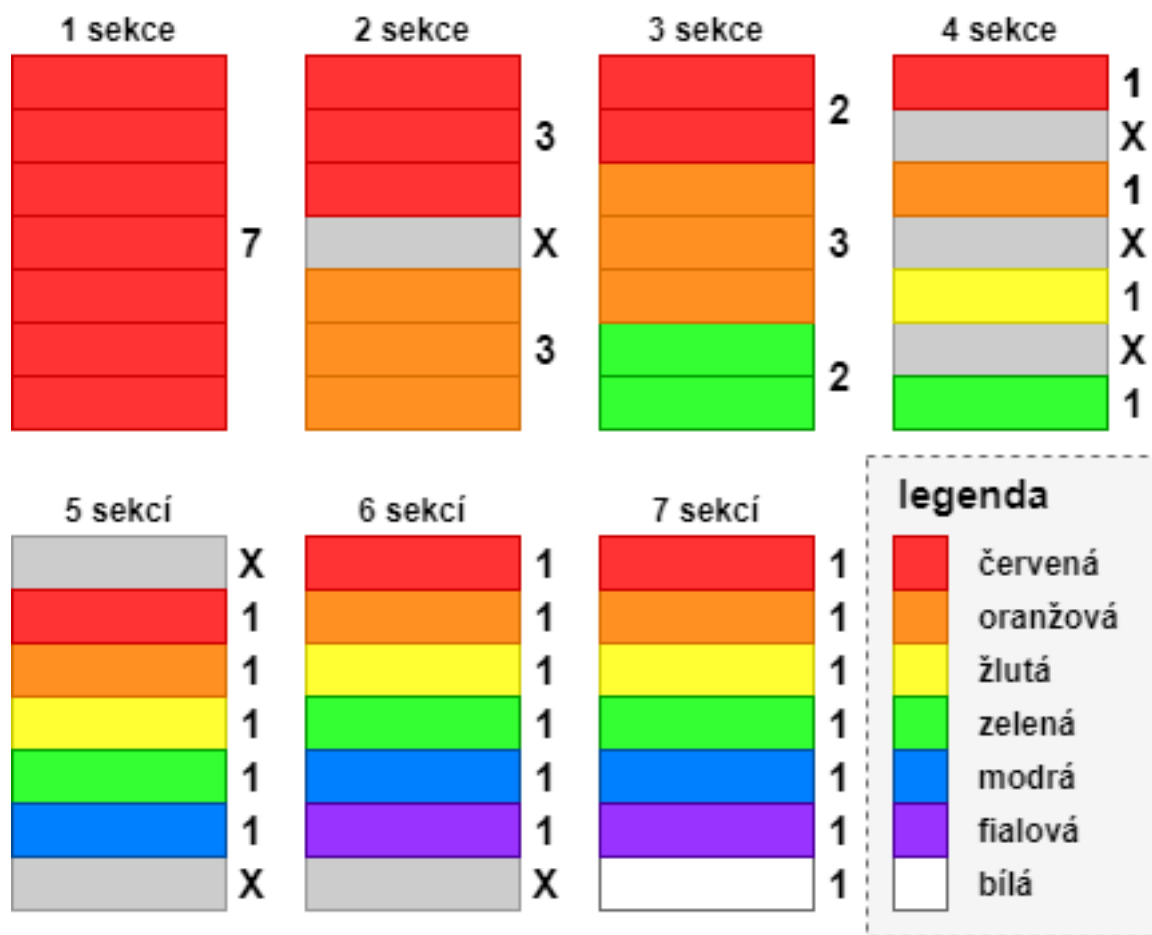
Barvy světla konkrétních stavů lze volit z **65535 odstínů** (zápisem do 16-bit holding registru) oproti běžným signalizačním jednotkám, kde je většinou pevně definováno max. 6 odstínů světla.

V každém módu je také možné nastavit si požadovanou **animaci světla** jednotlivých stavů. Operátor si může vybrat z těchto animací: stálý svit, blikání, rotace vpravo/vlevo.

Podrobnou funkci jednotlivých holding registrů a datových typů použitých v datovém modelu se zabývám v sekci 6.3.1.

Modulový zobrazovací mód

Podstatou **modulového módu** je zajištění podobné funkčnosti jako u běžných signalizačních majáků. To znamená, že zobrazující displej je rozdělen do několika sekcí (modulů) podle požadovaného počtu zobrazovaných stavů o maximálním počtu 7 (viz obr. 4.1). Tento mód je rozdělen do dalších dvou režimů – stavový a manuální.



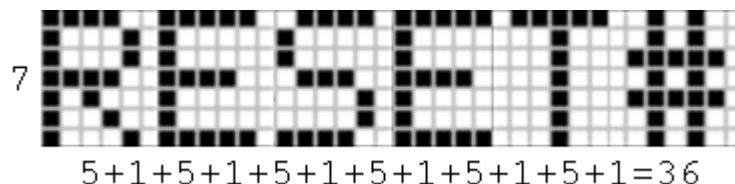
Obr. 4.1: Modulový zobrazovací mód

Jak ve **stavovém režimu**, tak i v **manuálním režimu** si lze zvolit barvu a animaci světla pro konkrétní moduly, jejichž počet závisí na zvolené konfiguraci, jak bylo již zmíněno v úvodu této sekce. Manuální režim navíc umožňuje operátorovi podle potřeby zapínání či vypínání konkrétních modulů.

Textový zobrazovací mód

Textový mód je nejzásadnějším rozdílem od běžných signalizačních majáků. Jeho podstatou je pochopitelně zobrazování textu, jehož font je naprogramován na šířku 5 pixelů a výšku 7 pixelů (viz obr. 4.2). Zobrazovaný text je zároveň podsvícen barevným pozadím, které odpovídá signalizovanému stavu.

V případě, že operátor zvolí ID stavu mimo předdefinovaný rozsah, může si nadefinovat vlastní název stavu (o maximální délce 6 znaků), který bude zároveň zobrazovaným textem na displeji.



Obr. 4.2: Font zobrazovaného textu na displeji

4.2 Koncepce HW

Na obr. 4.3 je zobrazen blokový diagram hardwarového řešení SSU. Jednotlivé bloky jsou rozděleny pomocí barev do skupin v závislosti na jejich hlavní funkci.

Modrý blok představuje samotný modul ESP32-WROOM-32, který je jádrem celé SSU a zajišťuje běh programu nahraného uvnitř něj.

Fialový blok je tvořen čipem LAN8710A, který se stará o zpracování komunikace zprostředkované přes Ethernet. Jedná se tedy o tzv. fyzickou vrstvu Ethernetu podporující rozhraní *Reduced Media Independent Interface* (RMII).

Červené bloky mají na starost vše, co souvisí s napájením. Spadá sem tedy především čip Si34061 a v nutném případě i externí zdroj. Ethernetový kabel s konektorem by bylo možné pochopitelně zařadit i do oblasti související s komunikací, tedy fialové. *Step-Down* regulátor se pak stará o převod napájecího napětí z 5 V na 3,3 V pro ESP32-WROOM-32 modul a tedy i pro čip LAN8710A.

Zelené bloky souvisí s *Universal Serial Bus* (USB) a *Universal Asynchronous Receiver/Transmitter* (UART) komunikací, pomocí kterých je ESP32-WROOM-32 modul programovatelný.

Žluté bloky označují periferie SSU, jimiž jsou digitální LED pásy WS2812B (displej), digitální teploměr DS18B20 a ventilátor pro zabránění přehřátí zařízení.

Všechny zvolené komponenty vycházejí z ověřených oficiálních návrhů různých výrobců a při tvorbě koncepce SSU **byly konzultovány s vedoucím práce.**

4.2.1 Popis funkce komponent v blokovém diagramu

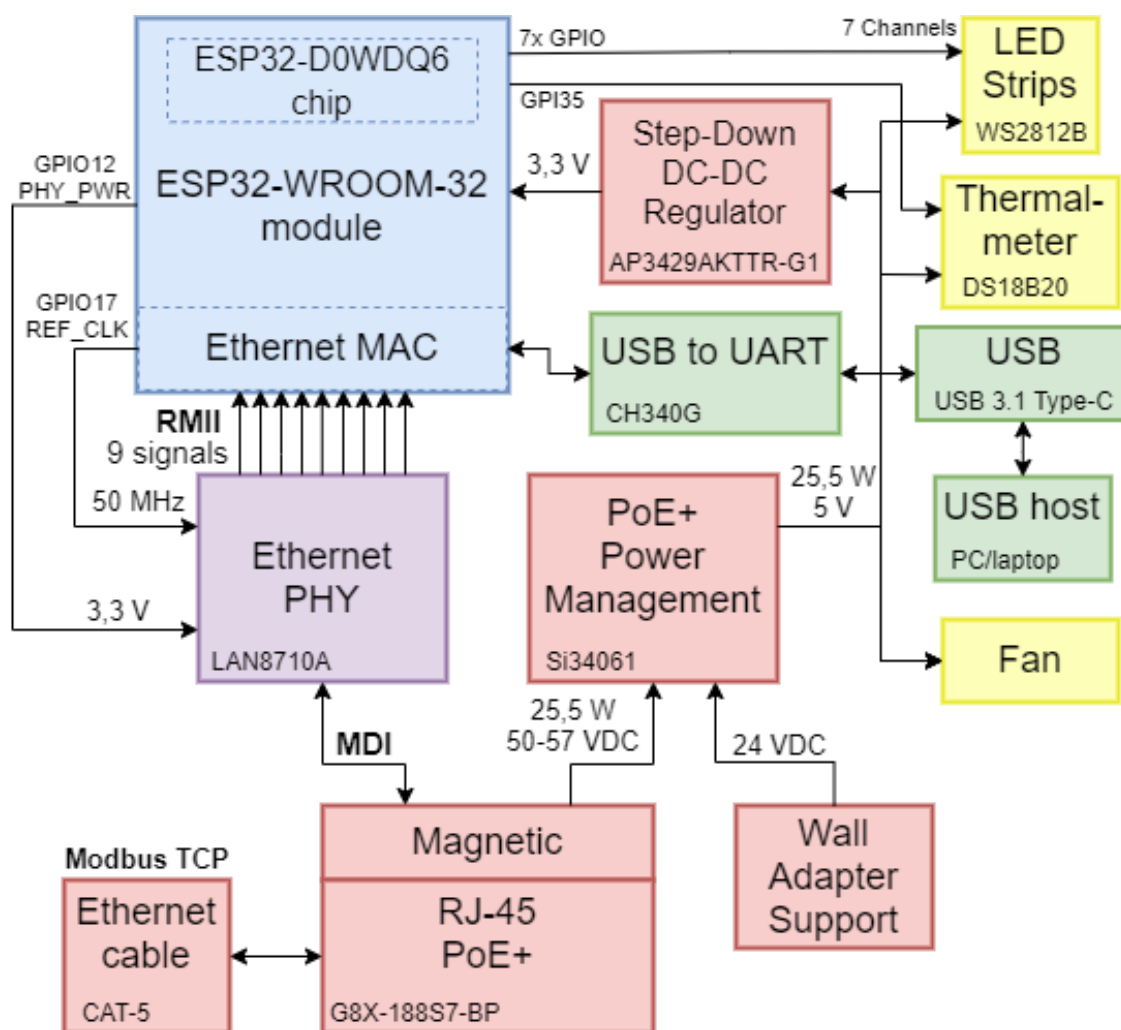
Následující sekce popisují jednotlivé bloky konceptu uvedeného v diagramu na obr. 4.3.

Modul ESP32-WROOM-32

Výkonný programovatelný modul ESP32-WROOM-32 (viz obrázek 4.4¹) od firmy Espressif je srdcem Chytré signalizační jednotky. V jádru ESP32-WROOM-32 je čip ESP32-D0WDQ6 obsahující dvě individuálně říditelná jádra s nastavitelným časem

¹Převzato a upraveno z:

<<https://dl.espressif.com/dl/schematics/pictures/esp32-wroom-32-front-back.jpg>>



Obr. 4.3: Blokový diagram Chytré signalizační jednotky

(CPU clock) mezi 80 MHz až 240 MHz. Tento čip nabízí velkou sadu periférií (SD karta, UART, SPI, I^2C , LED PWM, Motor PWM, I^2S , GPIO, ADC, DAC...) a díky integrovaným Bluetooth, BLE a Wi-Fi je použitelný pro širokou škálu aplikací od domovních automatizací a chytrých budov, přes průmyslovou automatizaci, robotiku, zpracování zvuku a obrazu, po „nositelnou“ (*wearable*) elektroniku díky svému velmi malému odběru. Modul má vlastní anténu s výkonem 20 dBm [16] [17].

Ethernet PHY – LAN8710A

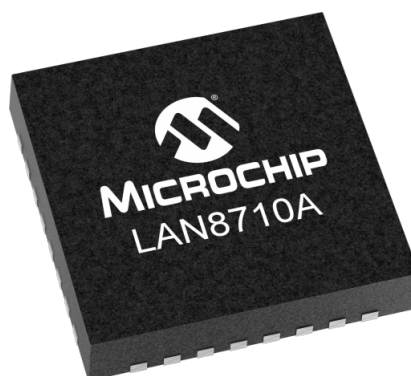
Jedná se o jednočipový ethernetový vysílač/přijímač (*transceiver*) fyzické vrstvy (*Physical Layer* (PHY)) s nízkou spotřebou, rychlostí podle standardů 10BASE-T/100BASE-TX a proměnným napětím na vstupech/výstupech (*Input/Output* (I/O)), které je v souladu se standardem IEEE 802.3-2005 [18]. Čip LAN8710A od firmy



Obr. 4.4: Modul ESP32-WROOM-32

Microchip lze vidět na obr. 4.5².

Podporuje komunikaci s Ethernet *Media Access Control* (MAC) prostřednictvím standardního rozhraní *Media Independent Interface* (MII)/RMII. Obsahuje plně-duplexní³ (*full-duplex*) transceiver a podporuje rychlosti 10 Mb/s (10BASE-T) a 100 Mb/s (100BASE-TX) vykonávání operací. Dále je v něm implementován HP Auto-MDIX, který zajišťuje automatické ustavení pracovního režimu (*auto-negotiation*) pro zajištění nejlepší možné rychlosti a „duplexity“. Podporuje tak použití jak přímých, tak i křížených LAN kabelů [18].



Obr. 4.5: Ethernet PHY – čip LAN8710A

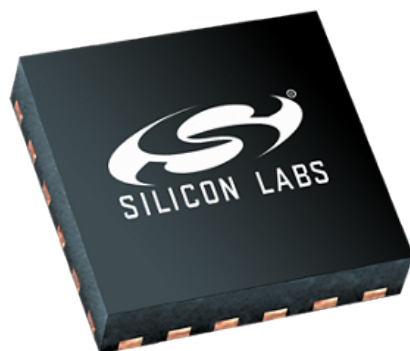
²Dostupné z:

<https://www.microchip.com/_images/products/medium/84a9a4cb154f85c0021ba09e39fbecda.png>

³Díky konstrukci Ethernetových kabelů, které používají ke komunikaci 2 (nebo dokonce 4) kroucené páry vodičů, lze snadno použít plný duplex, čímž mohou stanice současně přijímat i vysílat a rychlost obousměrné komunikace tak oproti polovičnímu duplexu vzroste na dvojnásobek. Zároveň je tak i zmenšena pravděpodobnost kolize [20].

PoE+ Power Management – Si34061

O správu napájení z rozhraní PoE+ PD se stará plně integrovaný čip Si34061 od firmy Silicon Labs s vysoce účinnými spínacími regulátory s režimem spánku a probuzení (viz obr. 4.6⁴).



Obr. 4.6: Čip Si34061 pro správu napájení pomocí PoE

Je kompatibilní se standardem IEEE 802.3. Jeho hlavní funkcí je převést vysoké napětí (44,0 – 57,0 VDC) dodané přes 10/100/1000BASE-T Ethernet na regulované nízkonapěťové výstupní napájení (5 VDC) [19].

Uvnitř jsou integrovány také požadované diodové můstky⁵ (*diode bridges*) a tlumič přechodného přepětí (*transient surge suppressor*), což umožňuje přímé připojení integrovaného obvodu (*Integrated Circuit* (IC)) ke konektoru Ethernet RJ-45. Integrovan je i výkonový spínací tranzistor *Field Effect Transistor* (FET)⁶ a všechny k němu přidružené funkce. Integrovaný DC-DC převodník obsahuje proudem řízený spínací regulátor, který podporuje topologii⁷ s izolovanou či neizolovanou zpětnou vazbou nebo topologii s využitím *buck* (*step-down*) převodníku. Frekvence spínání tohoto regulátoru je nastavitelná jednoduše přidáním rezistoru odpovídající hodnoty. Na sekundární straně hlavního izolačního transformátoru je možné přidat externí synchronizovaný FET pro dosažení lepší efektivity převodu napětí [19].

Připojení k PSE je během spánku udržováno volitelně automatizovaným signálem MPS. Čip plně podporuje specifikaci IEEE 802.3at, tedy jedno- či dvou-událostní

⁴Dostupné z:

<<https://siliconlabs-h.assetsadobe.com/is/image/content/dam/siliconlabs/images/products/chip-packages/qfn24.png>>

⁵Vzhledem k poměrně velkému odběru napájených LED pásků musí být implementovány 2 externí Schottkyho můstky (*Schottky diode bridge*) a interní můstky tak nejsou použity.

⁶Stejně tak tento výkonový spínací FET musí být doplněn o externí spínací FET vzhledem k potřebnému spínanému výkonu.

⁷Pro realizaci Chytré signalizační jednotky jsem zvolil izolovanou zpětnovazební topologii, která zajišťuje největší bezpečnost proti přepětí. Ta je vykoupena většími nároky na prostor na desce plošných spojů (DPS).

klasifikaci, čehož je dosaženo opět přidáním detekčního rezistoru odpovídající hodnoty pro danou výkonovou třídu. Interní spouštěcí obvody zajišťují dobře řízený pozvolný rozběh počátečního spuštění spínače *hotswap* i regulátoru napětí [19].

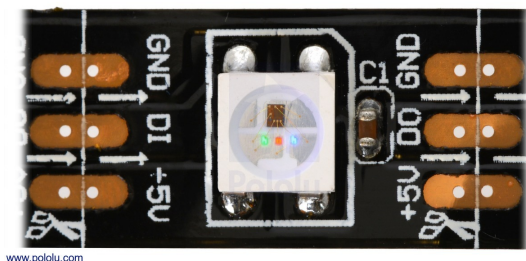
Je možné také připojit externí napájení v rozmezí 12 V až 57 V, které automaticky zastoupí funkci napájení přes PoE/PoE+.

Převodník z USB na UART – CH340G

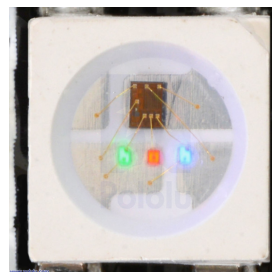
Pro převod ze sériové sběrnice USB na sériové rozhraní UART je možné použít například čip CH340G. Čip je kompatibilní s USB-3.1 Type-C a plně duplexní s vestavěnou vyrovnávací pamětí pro přenos a příjem typu *First In First Out* (FIFO), podporuje přenosovou rychlost komunikace od 50 b/s do 2 Mb/s [21].

Digitální LED pásy – WS2812B

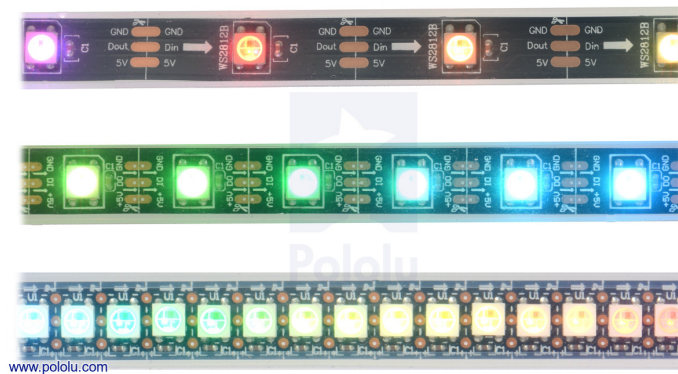
Digitální LED pásy WS2812B jsou ve světě velmi populární. Prodávají se v různých hustotách LED čipů s pouzdrem 5050 na 1 metr, např. 30, 60 nebo 144 (viz obr. 4.9). Jednou z funkcí SSU má být zobrazování 6ti znakového textu, čehož lze dosáhnout, při dodržení stanovených rozměrů SSU, pouze nejhustěji osazenou variantou LED pásků, tedy 144 na 1 metr. Obrázky níže jsou převzaty z <<https://www.pololu.com/product/2547>>.



Obr. 4.7: Osazený LED čip WS2812B



Obr. 4.8: Detail LED čipu WS2812B



Obr. 4.9: LED pásy WS2812B různých hustot osazení

Uvnitř čipu jsou 3 mikročipy schopné emitovat světlo jedné barvy (R/G/B). Jednotlivé LED čipy je dále možné individuálně adresovat díky miniaturnímu řadiči umístěném v každém z nich. Tyto řadiče také zajišťují rekonstrukci digitálního signálu při jeho průchodu – vstupní pin *Data Input* (DIN), výstupní pin *Data Output* (DOUT) (viz obr. 4.7). Každý čip (pixel) si odebere z tohoto vstupu svých 24 bitů, které odpovídají bitové posloupnosti příslušející světelným složkám G, R, B (viz tab. 4.1), a zbylé pošle rekonstruované následujícímu pixelu v kaskádovém zapojení. Tím je dosaženo libovolné délky LED pásku, která už je tak omezena pouze poklesem napájecího napětí *Voltage Drain Drain* (VDD) s rostoucí délkou. V našem případě máme kaskádově zapojeno pouze 36 LED čipů v každém samostatně napájené kanálu a obnova napájecího napětí tak není potřeba. Přesto je DPS navržena tak, aby byly jednotlivé LED pásy napájeny jak na začátku, tak i na konci.

Tab. 4.1: Složení 24bitových dat pro LED čip WS2812B, převzato a upraveno z [23]

G7 G6 G5 G4 G3 G2 G1 G0	R7 R6 R5 R4 R3 R2 R1 R0	B7 B6 B5 B4 B3 B2 B1 B0
-------------------------	-------------------------	-------------------------

Poznámka: Přenos dat probíhá v pořadí GRB, významný bit jako první.

Digitální teploměr – DS18B20

Digitální teploměr DS18B20 umožňuje měření teploty s programovatelným rozlišením od 9ti do 12ti bitů v rozmezí od $-55\text{ }^{\circ}\text{C}$ do $+125\text{ }^{\circ}\text{C}$, přičemž v rozmezí od $-10\text{ }^{\circ}\text{C}$ do $+85\text{ }^{\circ}\text{C}$ s přesností $\pm 0,5\text{ }^{\circ}\text{C}$.

Dále disponuje funkcí alarmu a uživatelem programovatelnými dolními a horními úrovněmi spuštění (*trigger points*). Komunikaci zprostředkovává po sběrnici 1-Wire, která ze své podstaty vyžaduje pro komunikaci s centrálním mikroprocesorem pouze jednu datovou linku (a zem). Navíc umí získávat energii přímo z datové linky (*parasite power*), čímž eliminuje potřebu externího napájení.

Každý DS18B20 má jedinečný 64-bitový sériový kód, který umožňuje fungování více těchto čipů na stejné 1-Wire sběrnici. Není tedy problém použít jeden mikroprocesor k ovládání mnoha distribuovaných DS18B20.

Mezi běžné aplikace patří například systémy monitorování teploty uvnitř budov, zařízení nebo strojů.

V designu SSU je použita varianta s 8-pinovým pouzdrem *Small Outline* (SO).

5 Realizace Chytré signalizační jednotky

V následujících podkapitolách se zabývám podrobným popisem postupu při návrhu a následné realizaci desky plošných spojů (DPS) pro SSU.

5.1 Návrh desky plošných spojů (DPS)

Kompletní návrh elektrického schéma i DPS probíhal ve velmi výkonném prostředí **Eagle verze 9.6.2** od firmy Autodesk, jehož vývoj má už více jak 30 let dlouhou historii.

Vzhledem k mým dosavadním zkušenostem v navrhování DPS v prostředí Eagle jsem si, před samotným vypracováváním bakalářské práce, musel nastudovat spoustu informací, které je vhodné (ba dokonce nutné) při návrhu dodržovat. K tomuto účelu jsem využil skvěle zpracovaná skripta pro kurz BPC-KEZ (Konstrukce el. zařízení).

Prvním krokem při návrhu DPS bylo potřeba si určit její **tvar a rozměry**. Jak už jsem zmínil přibližné rozměry v podkapitole 1.1.5, finální výška desky je 105 mm a šířka (průměr pro výpočet vnitřního obvodu) 67 mm.

Šířku měděných cest jsem počítal především podle velikosti elektrického proudu (**proudového zatížení**), který by měl být schopen projít bez výrazného ohřívání DPS. Takové ohřátí desky by totiž mohlo zapříčinit poměrně výrazné pnutí, které by mohlo vést i k utržení *Surface Mount Device* (SMD) součástek orientovaných ve směru pnutí. Dalším parametrem pro výpočet šířky cest byla samotná tloušťka měděné vrstvy DPS, která je standardně 0,035 mm (naš případ) nebo 0,075 mm. Volitelnými (upřesňujícími) parametry jsou dále například délka cest, s níž roste úbytek napětí, okolní teplota nebo hodnota vzrůstu teploty. K těmto výpočtům jsem využil online kalkulator¹, přičemž jsem vždy přidal ještě dostatečně velkou rezervu. Níže je příklad zadaných hodnot pro výpočet požadované šířky cesty.

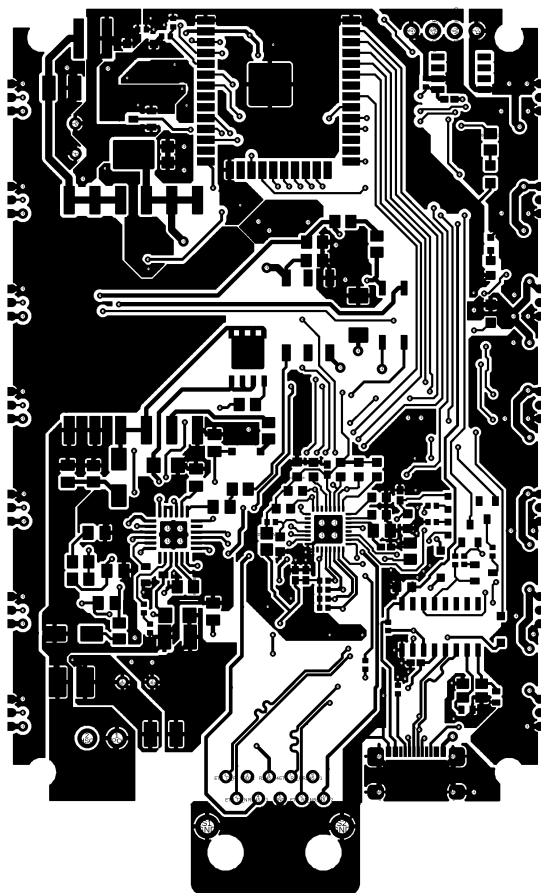
Proud:	0,6 A	Teplota okolí:	35 °C
Tloušťka mědi:	0,035 mm	Vzrůst teploty:	20 °C
		Délka cesty:	50 mm

Požadovaná šířka cesty (vnější vrstvy): **0,0975 mm**

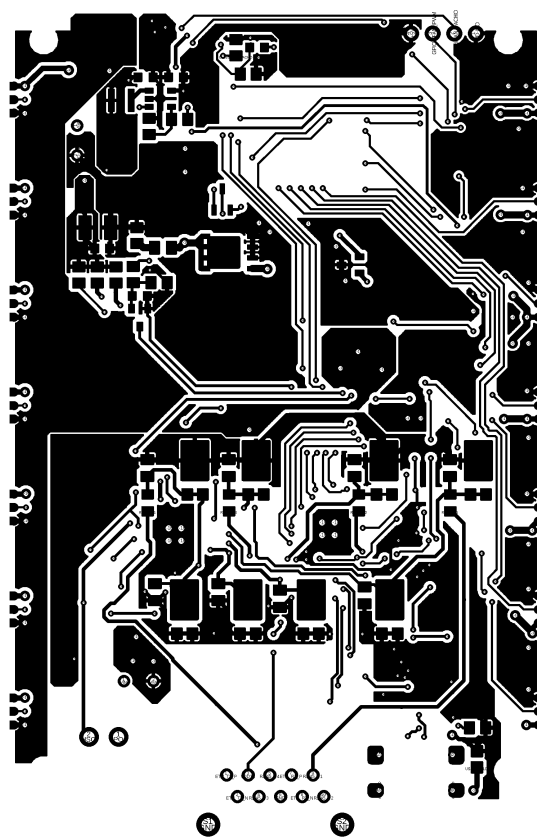
Nejlepšího rozvodu proudu, nízkého elektrického odporu a navíc mnohem lepšího odvodu tepla jsem dosáhl pomocí rozlité mědi – **polygonů**.

Pro případ, že by se nepovedlo zprovoznit část, která spravuje napájení pomocí PoE (vzhledem ke složitosti této sekce), přidal jsem na horní i spodní vrstvu velké plochy (k levému okraji desky z pohledu na vrchní vrstvu), kam je možné připojit napájecí vodiče, a připojit tak napětí 5 V. To lze dobře vidět na obrázcích 5.1 a 5.2.

¹Převzato a upraveno z: <<http://circuitcalculator.com/wordpress/?p=25/>>



Obr. 5.1: Vrchní vrstva DPS

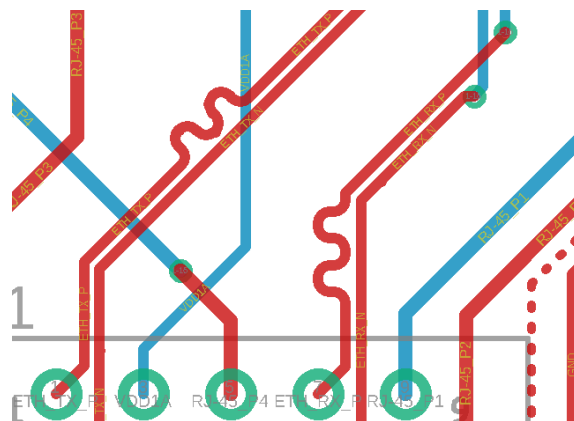


Obr. 5.2: Spodní vrstva DPS

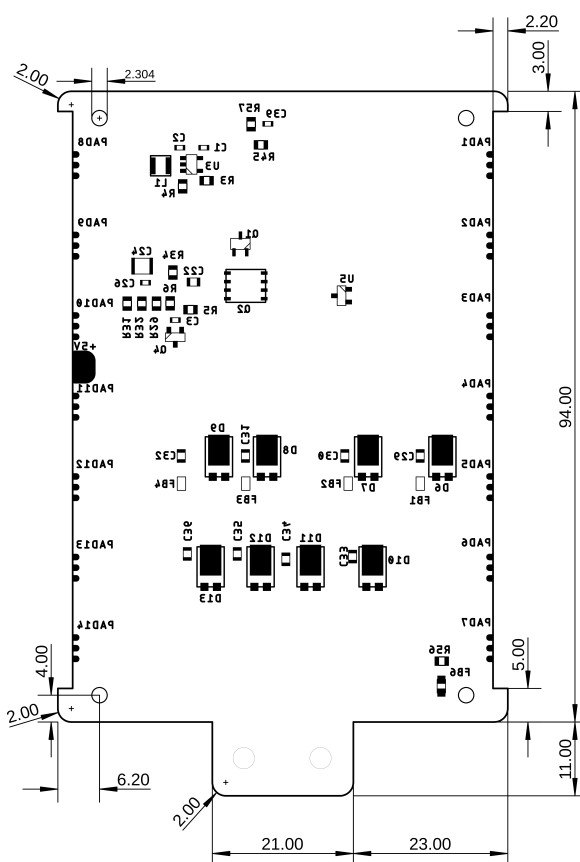
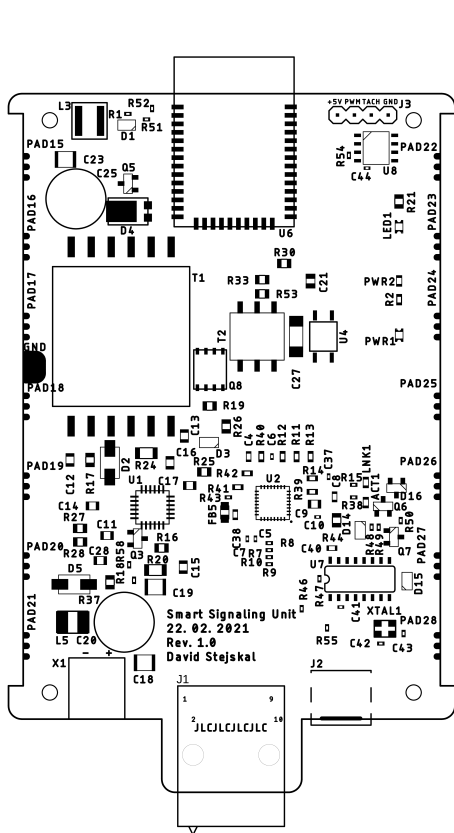
Pro dodržení stálých vzájemných elektrických vlastností signálových **diferenciálních párů** USB_D P/N, ETH_TX P/N a ETH_RX P/N jsem využil velmi šikovného nástroje *Route a Differential Pair*. Aby byla dodržena i ideálně stejná délka vodičů v rámci jednoho páru, použil jsem nástroj *Meander*. Využití obou těchto nástrojů při návrhu DPS si lze všimnout na obr. 5.3. Pro dodržení stálých vzájemných elektrických vlastností signálových **diferenciálních párů** USB_D P/N, ETH_TX P/N a ETH_RX P/N jsem využil velmi šikovného nástroje *Route a Differential Pair*. Aby byla dodržena i ideálně stejná délka vodičů v rámci jednoho páru, použil jsem nástroj *Meander*. Využití obou těchto nástrojů při návrhu DPS si lze všimnout na obr. 5.3.

Na obrázcích 5.4 a 5.5 je zobrazeno rozmístění komponent na finálním návrhu DPS. Dále si pak lze všimnout měděných ploch horní a spodní vrstvy DPS na obr. 5.1 a 5.2, které bylo nutné zaslat v příslušných souborech výrobci společně s dalšími výrobními podklady – ty lze vygenerovat ze souboru *Smart_Signaling_Unit.brd* (viz elektronická příloha).

Na obr. 5.6 si lze prohlédnout 3D model DPS vytvořený pomocí softwarového ná-



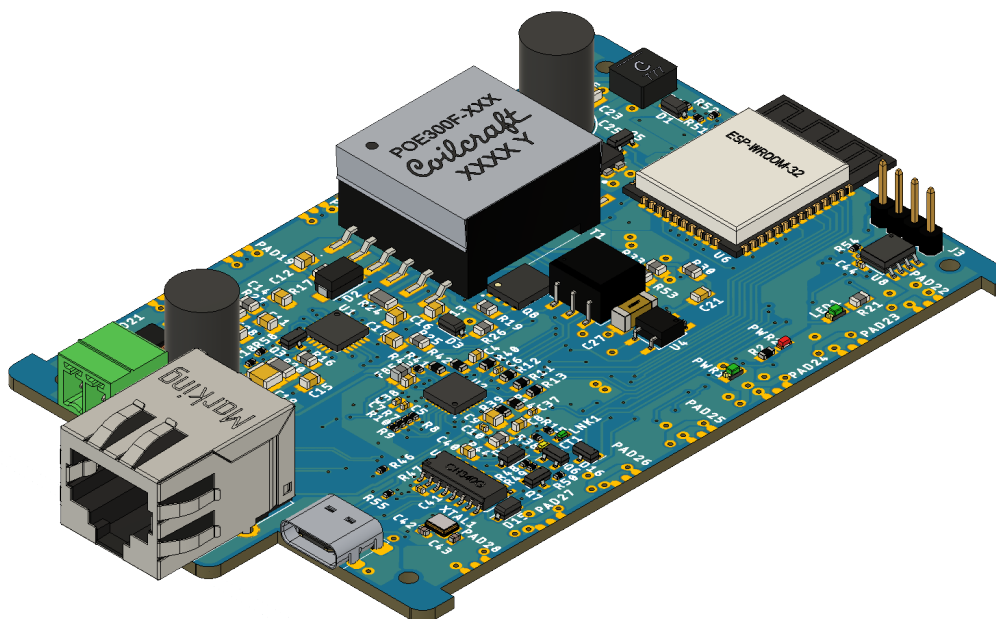
Obr. 5.3: Příklad využití nástrojů Route a Differential Pair a Meander



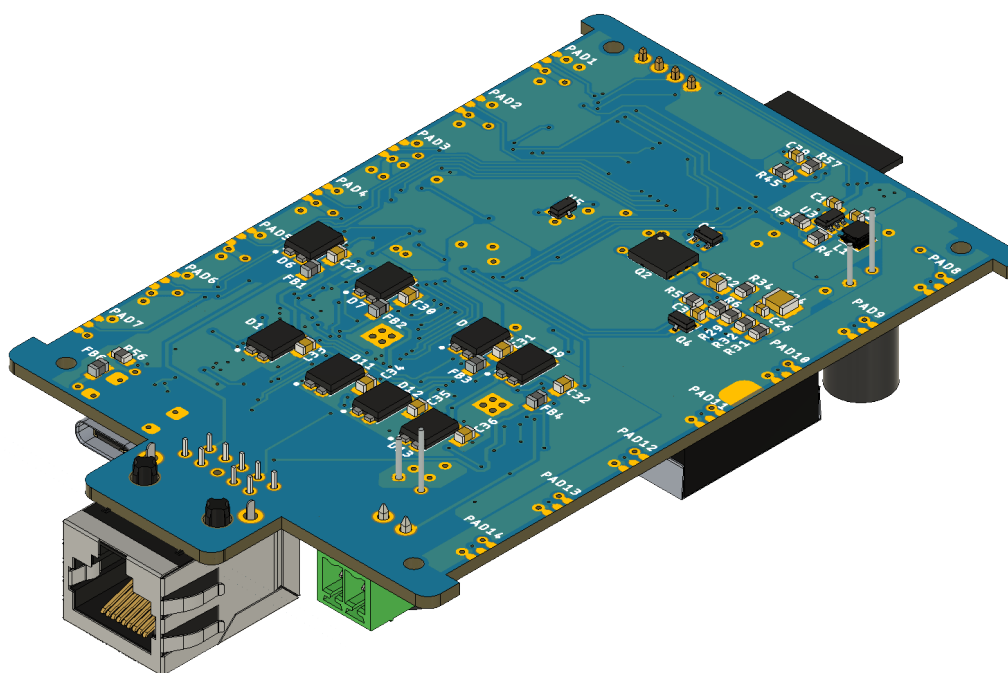
Obr. 5.4: Rozmístění komponent na
vrchní vrstvě DPS

Obr. 5.5: Rozmístění komponent na spodní
vrstvě DPS, rozměry

stroje **Fusion 360**, do kterého je možné vyexportovat návrh z prostředí Eagle a dále s ním pracovat – například tento model zakomponovat do většího celku (sestavy).



Obr. 5.6: 3D model DPS – vrchní vrstva



Obr. 5.7: 3D model DPS – spodní vrstva

5.1.1 Proudové zatížení

Při výpočtu proudového zatížení SSU je potřeba zohlednit především odběr řízených digitálních LED pásků (viz 4.2.1), protože v porovnání s ním je odběr řídicí elektroniky zanedbatelný.

V tabulce 3.1 je uveden **maximální výkon 25,5 W** a velikost **napětí** na napájení zařízení **50 V**, který je PoE+ schopno dodat v nejhorším případě, tedy hraničí-li délka ethernetového kabelu s maximální doporučenou délkou 100 m a úbytek napětí je tak největší. Tento dodaný výkon při napětí 50 V je dále transformován pomocí čipu Si34061 a několika dalších komponent na 5 V. Budeme-li tedy uvažovat nejhorší případ, který může nastat, a zmíněný převod, následujícím vztahem vypočítáme maximální odeberatelný proud:

$$I_{max} = \frac{P}{U}. \quad [A] \quad (5.1)$$

Tedy:

$$I_{max} = \frac{25,5}{5} = 5,1 \text{ A}.$$

Dále jsem potřeboval zjistit odebíraný proud jedné LED při plném svitu. Tuto informaci jsem z datového listu k LED čipu WS2812B bohužel nezjistil. Tento čip je ale velmi rozšířený, takže mnoho uživatelů již provedlo měření a přestože se jejich naměřené hodnoty mírně lišily, nejčastěji se vyskytovala hodnota proudu 17,5 mA při plném svitu jedné barvy, tzn. 52,5 mA pro plný svit všech tří barev (ve výsledku bílá barva). Tuto informaci jsem později sám ověřil měřením.

Celá zobrazovací plocha SSU se skládá ze sedmi kanálů (řádků pod sebou), přičemž v každém řádku je 36 LED čipů (pixelů). Celkem jich tedy je $7 \times 36 = 252$. Má-li majáček svítit všemi pixely s největší intenzitou např. zelenou barvou, znamenalo by to proudový odběr $I_{green,max} = 252 \times 0,0175 = 4,41 \text{ A}$, takže do maximálního odběru bychom se ještě s rezervou vešli. Pro zelené světlo je ale aktivní pouze G složka a odběr tím pádem není tak veliký.

Uvažujme tentokrát jantarovou (*amber*) barvu – RGB(255, 191, 0). Tato barva je u signalizačních majáčků velmi obvyklá. Odběr při takovém světle a plné intenzitě svitu lze spočítat následovně:

$$\begin{aligned} I_{amber} &= n \times I_m \times \left(\frac{r}{255} + \frac{g}{255} + \frac{b}{255} \right) \quad [A] \quad (5.2) \\ I_{amber} &= 252 \times 0,0175 \times \left(\frac{255}{255} + \frac{191}{255} + \frac{0}{255} \right) \\ I_{amber} &= 7,713 \text{ A}, \end{aligned}$$

kde n je počet svítících pixelů, I_m proud čipu při plném svitu jedné barvy a r, g, b hodnoty jednotlivých složek světla v rozsahu 0 – 255.

Z předešlého výpočtu je jasné vidět, že pro tuto aplikaci jsme při napájení přes PoE+ výkonově poměrně výrazně limitováni. Proto je **nutné realizovat programové omezení intenzity svitu pixelů**.

Zajímá mě ještě proudový odběr v případě, kdy celý maják svítí plnou intenzitou např. žlutou barvou – RGB(255, 255, 0) a zároveň je zobrazován text o čtyřech

znacích (např. stav HELD). Provedl jsem tedy následující výpočet, který je velmi podobný předešlému:

$$I_{yellow,txt} = (n - n_{off}) \times I_m \times \left(\frac{r}{255} + \frac{g}{255} + \frac{b}{255} \right) \quad [A] \quad (5.3)$$

$$I_{yellow,txt} = (252 - 62) \times 0,0175 \times \left(\frac{255}{255} + \frac{255}{255} + \frac{0}{255} \right)$$

$$I_{yellow,txt} = 6,65 \text{ A},$$

kde n_{off} je počet nesvítících pixelů – ostatní symboly jsou totožné s předešlým výpočtem.

I přesto, že by nesvítilo 62 pixelů, proudový odběr je 6,65 A.

Úplně nejhorší případ by pochopitelně nastal při svitu bílého světla při maximální intenzitě, tedy RGB(255, 255, 255) a bez zobrazovaného textu. V takovém případě by byl odebíraný proud $I_{clear,max} = 13,23 \text{ A}$.

5.1.2 Opravy desky plošných spojů

Po vyrobení první verze DPS jsme objevili několik chyb v návrhu. Nutné opravy pro zajištění správné funkčnosti desky a jejich řešení jsou podrobně popsány dále.

Úroveň signálu DIN LED pásků WS2812B

V tomto případě došlo z mé strany ke špatnému pochopení datového listu k LED čipům WS2812B, kde bylo uvedeno, že signál z digitálního výstupu (*Digital Output* (DO)) modulu ESP32-WROOM-32 do digitálního vstupu (DIN) LED čipu má být **minimálně** $0,7 \times VDD$. Při pročitání datového listu jsem tuto informaci správně pochopil jako $0,7 \times 5 \text{ V} = \mathbf{3,5 \text{ V}}$, tedy za VDD jsem považoval napájecí napětí LED pásky. Už jsem si ale neuvědomil, že napěťová úroveň DO modulu je **3,3 V**, nikoli 5 V. I přesto jsem zkusil LED pásek s $VDD = 5 \text{ V}$ ovládat digitálním výstupem 3,3 V a kupodivu bez problému fungoval.

Abychom ale dodrželi doporučené hodnoty v datovém listu, rozhodli jsme se přidat Schottkyho diodu PDS1040-13 od společnosti DIODES Inc. v propustném směru mezi napájecí větev +5 V a potenciál, na nějž připojuji piny VDD LED pásků. Na diodě tak vznikne úbytek napětí 0,45 V až 0,49 V (pro $I_F = 8 \text{ A}$, $T_S = +25 \text{ °C}$). Díky tomuto úbytku jsou LED pásy napájeny napětím cca 4,55 V a na DIN tak stačí přivést alespoň $0,7 \times 4,55 \text{ V} = 3,185 \text{ V}$ – to už je méně než 3,3 V.

Musel jsem použít 2 diody, jednu pro levou stranu a druhou pro pravou stranu DPS, kde se nachází „troj-plošky“, ke kterým jsou jednotlivé LED pásy (kanály 1–7) připájeny.

Špatná rozteč a velikost plošek pro připájení LED pásků

Rozteč plošek na LED páscích se od různých výrobců nepatrně liší a pohybuje se okolo 8-9 mm mezi krajními ploškami pásku, přičemž třetí ploška je přesně uprostřed, tzn. 4-4,5 mm od krajní.

Toto rozmezí by rozhodně nebyl problém při kompletaci. Při tvorbě DPS jsem udělal tu chybu, že jsem rozteč těchto plošek, při jejich návrhu v prostředí Eagle, volil podle rozteče pinů LED čipu, nikoli podle LED pásku, a tak je tato rozteč na DPS pouze 3 mm.

Nutná změna pinu pro digitální teploměr DS18B20

Při návrhu desky jsem si neuvědomil skutečnost, že digitální teploměr DS18B20 potřebuje pro svou správnou funkčnost připojení k *General Purpose Input/Output* (GPIO), který navíc obsahuje *Analog-to-Digital Converter* (ADC), nikoli pouze k *General Purpose Input* (GPI) (v návrhu desky GPI35). Těmto požadavkům odpovídá hned vedlejší pin č. 8 (GPIO33), takže oprava spočívala v přerušení stávající cesty k pinu č. 7 a napojení pomocí izolovaného vodiče na pin č. 8.

5.2 Bezpečnost zařízení

Bezpečnost by měla být při návrhu jakéhokoli zařízení vždy na prvním místě. V elektrických obvodech se jí běžně dosahuje bezpečnostními prvky jako jsou různé typy izolací a zábran, vratné či nevratné pojistky, galvanická oddělení, ventilátory nebo chladiče pro zabránění přehřátí, atd.

Níže je uvedeno několik bezpečnostních prvků, které jsou v Chytré signalizační jednotce implementovány.

- Konektor RJ-45 uvnitř obsahuje oddělovací transformátory pro zajištění galvanického oddělení.
- Uvnitř čipu Si34061 (na vstupu mezi piny Vpos a GND) je transil (TVS) na 100 V, který zabraňuje poškození zařízení přepětím na vstupu. Navíc je uvnitř integrována i ochrana proti přetížení a zkratu výstupu.
- Chytrá signalizační jednotka je vybavena digitálním teploměrem a ventilátorem pro zabránění přehřátí vzhledem k velikosti odebíraného proudu digitálními LED pásky.
- Zařízení dále disponuje tlumivkami uvnitř konektoru RJ-45 a několika dalšími indukčnostmi uvnitř obvodu k potlačení vlivů elektromagnetické indukce (*Electromagnetic Induction* (EMI)).

- Výstupní napětí 5 V je galvanicky odděleno pomocí optočlenu, transformátoru, DC-DC převodníku a vysokonapětového kondenzátoru, které jsou součástí zpětnovazebního (*flyback*) návrhu.

V první iteraci designu jsem opomněl ochranu v obvodech napájení pro jištění proti zkratu, které by mohlo mít za následek jak poškození zařízení, tak nežádoucí zatížení sítě, k níž je zařízení připojeno.

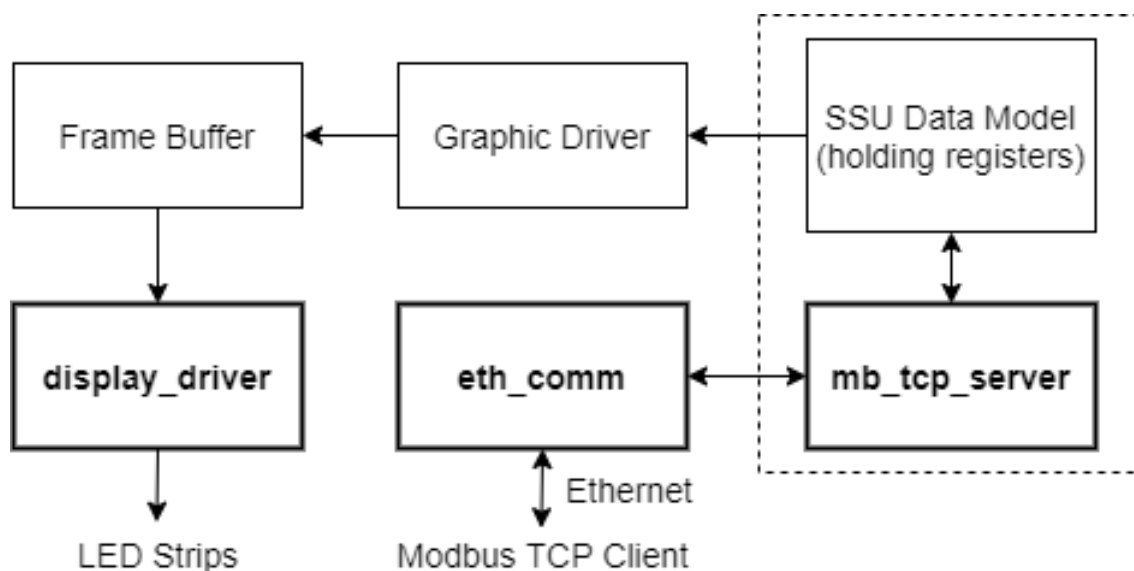
6 Návrh a implementace firmwaru SSU

V této kapitole podrobně popíšu koncepci *firmwaru* SSU.

Základ tvoří 3 „komponenty“ (viz obr. 6.1) zajišťující: komunikaci přes Ethernet (`eth_comm`), komunikaci pomocí Modbus TCP protokolu a implementaci Modbus serveru (`mb_tcp_server` společně s dalšími komponenty z frameworku ESP-IDF), a ovládání displeje složeného z digitálních LED pásků (`display_driver`).

„*SSU Data Model*“ představuje adresový prostor Modbus serveru (viz sekce 6.3.1) pro zápis/čtení Modbus klientů. Grafický řadič zapisuje do „*frame bufferu*“ (vyrovňovací paměť displeje), z něž řadič displeje průběžně načítá data a následně na základě těchto dat řídí displej.

V této verzi firmwaru jsou frame buffer s grafickým řadičem součástí řadiče displeje, ale do budoucna by bylo vhodné alespoň grafický řadič oddělit pro lepší logickou strukturovanost kódu.



Obr. 6.1: Blokový diagram firmwaru SSU

6.1 Softwarové nástroje

V této podkapitole popisuji všechny softwarové nástroje, které jsem použil při realizaci firmwaru pro SSU. Výběr je poměrně obrovský a vždy záleží na preferencích programátora nebo na požadavcích zadavatele úkolu. V mém případě jsem měl v tomto zcela volnou volbu a rozhodl jsem se pro následující.

6.1.1 ESP-IDF

Pro tvorbu firmwaru jsem použil oficiální *framework* od firmy Espressif nesoucí název *Espressif IoT Development Framework* (ESP-IDF) v4.4-dev-1183-g9d34a1cd4-dirty. Ten implementuje *Application Programming Interface* (API), což je v informatice označení pro programátorské rozhraní obsahující sbírku funkcí, procedur, tříd, knihoven atd., které může programátor využívat a ušetřit si tak spoustu práce a času. Celý tento framework je psaný v jazyce C.

6.1.2 Visual Studio Code

Kód jsem psal ve vývojovém prostředí, které je programátorům velmi známé a oblíbené – *Visual Studio Code* (VSCode) verze 1.56.2. Jedná se o volně dostupný (*open-source*) editor podporovaný na operačních systémech Windows, Linux i MacOS. Sám o sobě je to poměrně *light-weight* program, který má ale obrovskou podporu ve formě rozšíření (*extensions*) a každý programátor si ho tak může přizpůsobit svým potřebám a zvykům.

Pro psaní firmwaru jsem si vybral programovací jazyk C++, a to především proto, že mě provází po celou dobu bakalářského studia (beru-li v potaz jeho těsnou návaznost na jazyk C), a je mi tak ze všech programovacích jazyků nejbližší. Jazyk C++ je navíc velmi často používaný i v praxi díky své rychlosti (oproti konkurujícím programovacím jazykům).

6.1.3 CMake

CMake je rozšiřitelný, open-source systém, který spravuje, programátorům dobře známý, *build* proces způsobem nezávislým na překladači (kompilátoru). Na rozdíl od mnoha systémů pro různé platformy je CMake navržen pro použití ve spojení s nativním build prostředím. Poměrně jednoduché konfigurační soubory (nazývané CMakeLists.txt), umístěné v každém zdrojovém adresáři, se používají ke generování standardních build souborů (např. Makefiles pro Unix a projekty či pracovní prostory (*workspaces*) ve Windows MSVC), které se následně používají obvyklým způsobem.

Na tomto „mocném“ nástroji je postaven celý framework ESP-IDF, proto byl využit i při implementaci firmwaru pro SSU, konkrétně verze 3.16.4.

6.1.4 ModbusTool

ModbusTool je open-source nástroj pro testování Modbus *User Datagram Protocol* (UDP)/TCP/RTU s funkcemi pro import/export dat k zápisu/čtení. Nástroj pro simulaci Modbus Mastera (klienta) implementuje základní standardní funkce

Modbusu (např. *Read holding register*, *Write single register*, *Write multiple register* ad.). Podporuje několik formátů pro zobrazení hodnot (LED, *Hex*, *Binary*, ad.).

Práce s ním je poměrně intuitivní, takže nebyla k pochopení jeho funkce potřeba ani dokumentace. Pro otestování implementovaného Modbus serveru bylo zapotřebí nastavit následující hodnoty: Mode: TCP, Port: 502, IP Address: 192.168.0.144 (záleží na routerem přidělené adrese), Slave ID: 1. Hodnota Start Address určuje, od jaké adresy má klient zapisovat/číst a hodnota Size, s kolika registry má pracovat od adresy Start Address.

Na spodní části obrazovky se nachází konzolové okno *Communication Log*, kde klient vypisuje odchozí (*Transmit* (TX)) a příchozí (*Receive* (RX)) zprávy ve formátu Application Data Unit (viz 3.7). Tyto zprávy lze „přeložit“ například pomocí online analyzátoru (*parser*). Na obr. 6.2 je ukázka takového překladu pomocí online nástroje „*Rapid SCADA Modbus Parser*“ pro ADU zprávu:

TX: 00 00 00 00 00 06 01 03 00 04 00 40.

Part of Data Package	Description	Value
00 00	Transaction identifier	0x0000 (0)
00 00	Protocol identifier	0 = MODBUS protocol
00 06	Length	0x0006 (6)
01	Unit identifier	0x01 (1)
03	Function code	0x03 (3) - Read Holding Registers
00 04	Starting address	Physical: 0x0004 (4) Logical: 0x0005 (5)
00 40	Quantity	0x0040 (64)

Obr. 6.2: Příklad překladu ADU zprávy pomocí online parseru

6.2 Komunikace přes Ethernet

První z hlavních částí firmwaru má na starost zprostředkování komunikace pomocí síťového protokolu Ethernet. SSU používá rozhraní Reduced Media Independent Interface (RMII), jehož funkce je zajištěna pomocí maximálně devíti vodičů. Dále je pro ethernetovou komunikaci využíván TCP/IP zásobník (*stack*).

Tato část programu využívá především funkce z knihovny `esp_eth.h`, která je součástí frameworku ESP-IDF. Na základě těchto funkcí (a volně dostupného příkladu (*example*)) byly implementovány funkce `eth_start()`, `eth_connect()`, `eth_stop()` a `eth_disconnect()`.

Ve výpisu 6.1 je úryvek zdrojového kódu, který zajišťuje vytvoření instance ethernetové podvrstvy řídící přístup na médium (MAC) a instance ethernetové fyzické vrstvy (PHY), a následně jejich konfiguraci.

Výpis 6.1: Vytvoření síťového rozhraní a instancí MAC i PHY s jejich konfigurací

```

71 ESP_ERROR_CHECK(esp_netif_init());
72 esp_netif_config_t netif_cfg = ESP_NETIF_DEFAULT_ETH();
73 s_esp_netif = esp_netif_new(&netif_cfg);
74
75 eth_mac_config_t eth_mac_config = ETH_MAC_DEFAULT_CONFIG();
76 eth_mac_config.smi_mdc_gpio_num = CONFIG_ETH_MDC_GPIO;
77 eth_mac_config.smi_mdio_gpio_num = CONFIG_ETH_MDIO_GPIO;
78 esp_eth_mac_t *s_eth_mac =
    esp_eth_mac_new_esp32(&eth_mac_config);
79
80 eth_phy_config_t eth_phy_config = ETH_PHY_DEFAULT_CONFIG();
81 eth_phy_config.phy_addr = CONFIG_ETH_PHY_ADDR;
82 eth_phy_config.reset_gpio_num = CONFIG_ETH_PHY_RST_GPIO;
83 gpio_pad_select_gpio(CONFIG_ETH_PHY_POWER_PIN);
84 gpio_set_direction(CONFIG_ETH_PHY_POWER_PIN,
    GPIO_MODE_OUTPUT);
85 gpio_set_level(CONFIG_ETH_PHY_POWER_PIN, 1);
86 esp_eth_phy_t *s_eth_phy =
    esp_eth_phy_new_lan8720(&eth_phy_config);
87
88 esp_eth_config_t eth_config = ETH_DEFAULT_CONFIG(s_eth_mac,
    s_eth_phy);

```

Na výpisu 6.2 je ukázána instalace ethernetového řadiče, který je následně připojen k již zmíněnému TCP/IP stacku. Nakonec je spuštěn konečný stavový automat *Finite-State Machine* (FSM) ethernetového řadiče.

Výpis 6.2: Instalace ethernetového řadiče a připojení k TCP/IP stacku

```

98 ESP_ERROR_CHECK(esp_eth_driver_install(&eth_config,
    &s_eth_handle));
99
100 /* Attach Ethernet driver to TCP/IP stack */
101 s_eth_glue = esp_eth_new_netif_glue(s_eth_handle);
102 ESP_ERROR_CHECK(esp_netif_attach(s_esp_netif, s_eth_glue));
103
104 /* Start Ethernet driver state machine */
105 ESP_ERROR_CHECK(esp_eth_start(s_eth_handle));

```

Ethernetový řadič pošle žádost síťovému zařízení, které je schopné přidělovat IP adresy (směrovač – *router*), k němuž musí být připojené pomocí ethernetového

kabelu (jedná-li se o kabelové spojení, nikoli bezdrátové). Po přidělení IP adresy je již vše v pořádku a přichází na řadu vytvoření Modbus TCP serveru, který je na TCP/IP stacku rovněž taky založen.

Na kompletní kód lze nahlédnout do elektronické přílohy.

6.3 Modbus TCP server

Tato kapitola se zabývá druhou z hlavních částí firmwaru – implementací Modbus TCP serveru. Pro ten bylo nejdůležitější vymyslet **datový model**, jež detailně popisují dále.

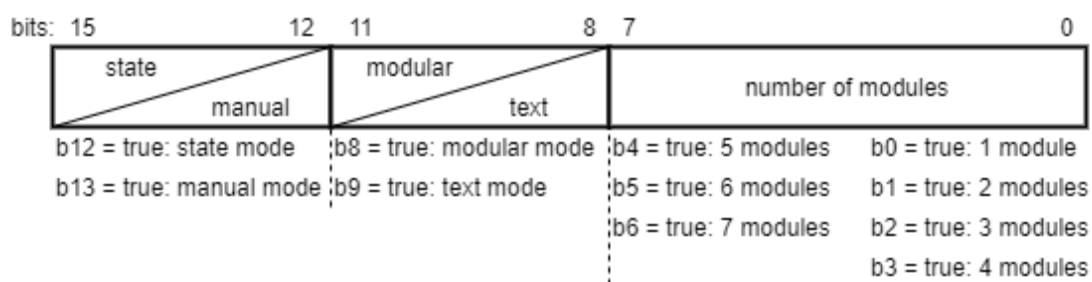
6.3.1 Datový model

Datový model definuje, jak jsou data ukládána v registrech Modbus serveru. V rámci SSU byly použity pouze uchovávací (holding) registry, k nimž mají Modbus klienti přístup jak pro čtení, tak pro zápis. **Zápisem** klienta do konkrétního registru tak lze **změnit odpovídající funkci SSU**. V následujících podsekcích popisují jednotlivé datové typy implementované v datovém modelu.

Kompletní datový model s konkrétními adresami a společně s datovými typy je součástí elektronické přílohy a do práce nebyl přidán kvůli své rozsáhlé velikosti.

CConfiguration_t

Strukturu datového typu CConfiguration_t, který je zároveň pouze jedním holding registrem, popisuje obr. 6.3.



Obr. 6.3: Konfigurační registr (CConfiguration_t) z datového modelu SSU

Prvních 7 bitů tedy slouží pro **nastavení konfigurace**, resp. do kolika modulů bude rozděleno 7 kanálů, které SSU obsahuje. Toto rozdělení je znázorněno na obr. 4.1 v sekci 4.1.1. Osmý a devátý bit nastavují, zda bude SSU zobrazovat **v modulovém nebo textovém módu**. Dvanáctý a třináctý bit jsou **přepínacími bity** pro modulový zobrazovací mód **mezi stavovým a manuálním režimem**.

Hodnoty registru v hexadecimální soustavě pro jednotlivá nastavení konfiguračního registru jsou uvedeny v tab. 6.1.

Tab. 6.1: Hodnoty konfiguračního registru

Název: CConfiguration_t	
Hodnota	Popis
0x0001	1 modul
0x0002	2 moduly
0x0004	3 moduly
0x0008	4 moduly
0x0010	5 moduly
0x0020	6 moduly
0x0040	7 moduly
0x0100	modulární mód
0x0200	textový mód
0x0800	stavový režim
0x1000	manuální režim

CState_t

Tabulka 6.2 popisuje význam hodnot registru pro volbu z množiny předdefinovaných stavů. Bude-li tedy požadavek na **nastavení názvu** např. *Aborted* **pro určitý stav**, registr CState_t pro daný stav bude mít hodnotu 0x0008.

Tab. 6.2: Hodnoty registru pro volbu názvu stavu

Název:		CState_t	
Hodnota	Popis	Hodnota	Popis
0x0000	Clearing	0x0009	Holding
0x0001	Stopped	0x0010	Held
0x0002	Starting	0x0011	Unholding
0x0003	Idle	0x0012	Suspending
0x0004	Suspended	0x0013	Unsuspending
0x0005	Execute	0x0014	Reseting
0x0006	Stopping	0x0015	Completing
0x0007	Aborting	0x0016	Complete
0x0008	Aborted		

CAnimation_t

Tabulka 6.3 popisuje význam jednotlivých bitů registru pro **volbu animace a případně délku doby rotace** pro konkrétní modul či stav.

Hodnota 0x0000 znamená, že modul nesvítí – toho lze využít např. při manuálním režimu, kdy je možné vypínat a zapínat jednotlivé moduly podle potřeby a rozsvítit jich tak klidně několik najednou.

Bity 4-15 slouží k nastavení doby v milisekundách potřebné na jeden oběh majáčku při aktivní animaci rotace, tzn. až 4095 ms.

Tab. 6.3: Hodnoty registru pro volbu animace

Název: CAnimation_t	
Hodnota	Popis
0x0000	vypnuto
0x0001	stálý svit
0x0002	blikání
0x0004	rotace doprava
0x0008	rotace doleva
0x000X	
...	čas rotace v milisekundách
0xFFFF	

CRgb_565_t

V tab. 6.4 je znázorněn význam jednotlivých bitů registru pro **zápis požadované barvy stavu**. Název registru (a jeho velikost) už trochu napovídá, že barva stavu musí být v holding registrech **uchována pomocí 16ti-bitového kódování**. Toto kódování typicky rozděluje skupinu bitů v poměru 5:6:5, tedy 5 bitů pro červenou složku, 6 bitů pro zelenou složku a 5 bitů pro modrou složku světla. Větší počet přidělených bitů pro zelenou složku není náhodný – lidské oko je na zelenou barvu nejcitlivější. Toto kódování je pak pomocí poměrně snadného algoritmu převedeno na kódování 8:8:8, které je pak následně posíláno do digitálních LED čipů na páscích.

6.4 Řadič displeje

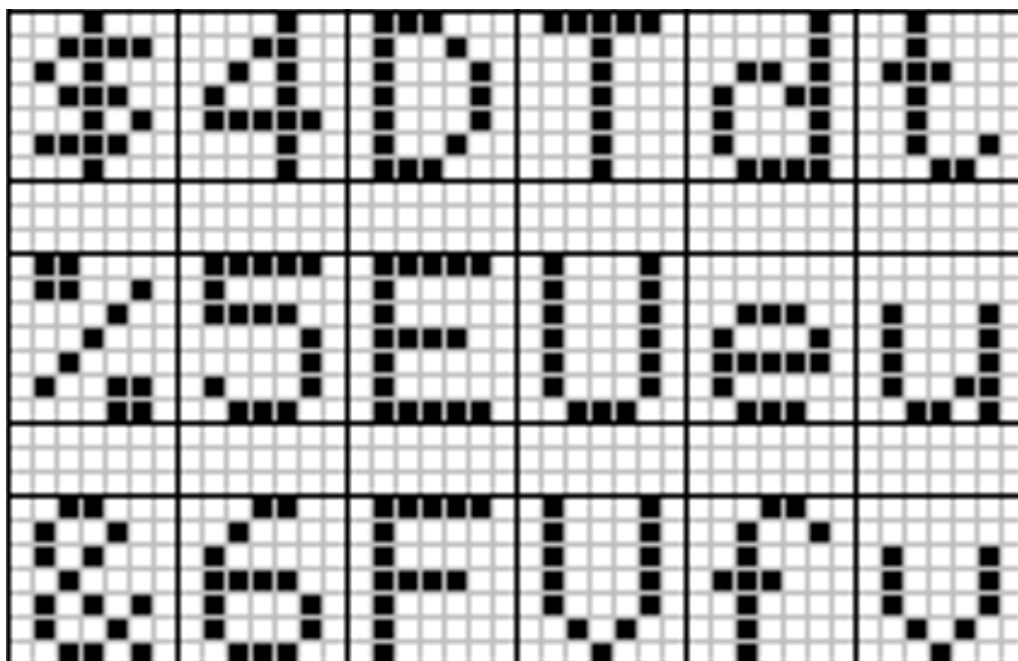
Jak již bylo zmíněno na začátku této kapitoly, řadič displeje v této verzi firmwaru zastupuje i funkci vyrovnávací paměti displeje a grafického řadiče. Grafický řadič by bylo vhodné oddělit jako samostatnou komponentu (podle terminologie CMake).

Tab. 6.4: Význam bitů registru pro nastavení barvy

Název:	CRgb_565_t
Hodnota	Popis
0b0000_0XXX_XXXX_XXXX ...	červená(4:0)
0b1111_1XXX_XXXX_XXXX ...	zelená(5:0)
0bXXXX_X000_000X_XXXX ...	modrá(4:0)
0bXXXX_XXXX_XXX0_0000 ...	
0bXXXX_XXXX_XXX1_1111	

Řadič displeje využívá knihovny `led_strip.h` psané v jazyce C, která byla v době tvorby firmwaru neoficiální součástí frameworku ESP-IDF. Ta obsahuje elementární funkce jako: `set_pixel()`, `refresh()`, `clear()`, `led_strip_init()` a několik dalších pro práci s LED pásky WS2812(B).

Pro správnou funkčnost zobrazování na displeji bylo nejdříve potřeba vymyslet a implementovat font textu o rozměru 5×7 pixelů. Na obr. 6.4 je ukázka z vytvořené ASCII tabulky. Inspiroval jsem se spoustou příkladů dostupných na internetu.



Obr. 6.4: Ukázka z implementované ASCII tabulky s fontem 5×7 pixelů

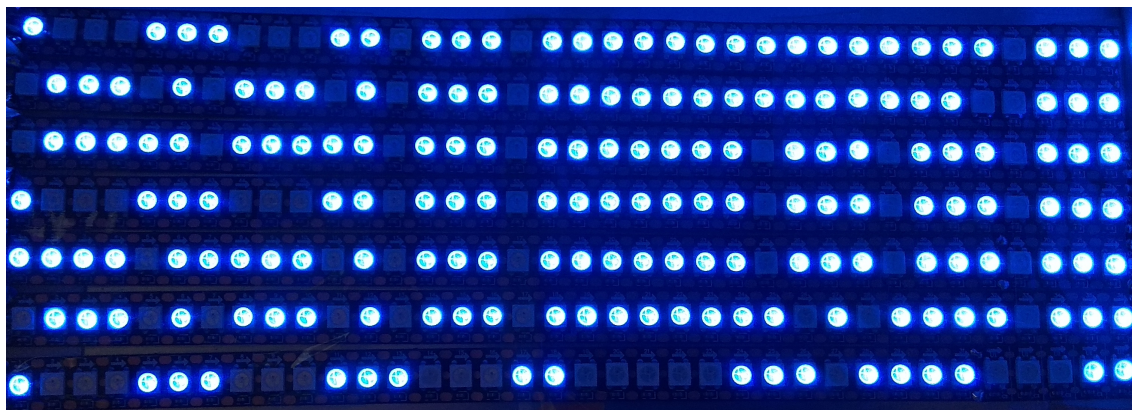
Takto vytvořenou tabulku jsem zapsal do 2-D pole `ascii_tab_5x7_[256]` [7] s prvky typu `uint8_t`. Znaky jsou do pole zapsané po řádcích, tzn. jednotlivé sloupce představují hexadecimální hodnoty příslušejících řádků daného znaku, přičemž každý řádek pole odpovídá jednomu znaku. Pro lepší představu přikládám krátký výpis 6.3.

Výpis 6.3: Ukázka fontu znaků 5x7 pixelů zapsaného v 2D poli

265	{0x0e, 0x02, 0x02, 0x02, 0x02, 0x02, 0x0e}, // 0x5d,]
266	{0x04, 0x0a, 0x11, 0x00, 0x00, 0x00, 0x00}, // 0x5e, ^
267	{0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f}, // 0x5f, _
268	{0x08, 0x04, 0x02, 0x00, 0x00, 0x00, 0x00}, // 0x60, ‘
269	{0x00, 0x00, 0x0e, 0x01, 0x0f, 0x11, 0x0f}, // 0x61, a
270	{0x10, 0x10, 0x16, 0x1c, 0x19, 0x11, 0x1e}, // 0x62, b
271	{0x00, 0x00, 0x0e, 0x10, 0x10, 0x10, 0x0e}, // 0x63, c

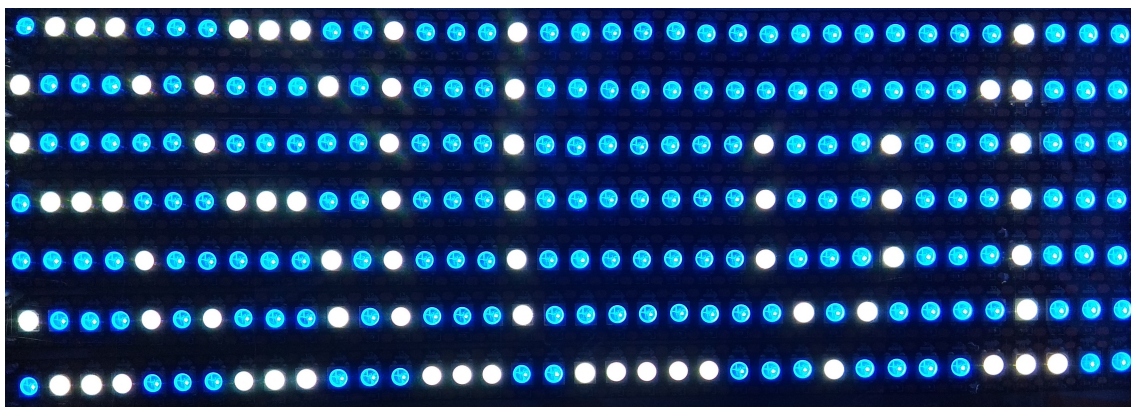
Následovala implementace plnění frame bufferu, což byla nejsložitější část řadiče displeje. Nejprve jsem vytvořil algoritmus pro nastavení masky, podle které se následně plnil frame buffer hodnotami nastavené barvy pozadí nebo jejího negativu v případě pixelů patřících textu.

Při prvním návrhu textového zobrazovacího módu byla myšlenka taková, že by pixely představující text neměly svítit – tedy svítilo by pouze barevné pozadí. Během testování této implementace bylo ale hned jasné, že bez rozsvícení pixelů představujících text nebude název stavu dobře čitelný (viz obr. 6.5).



Obr. 6.5: Zobrazování textu s vypnutými pixely pro text

Z tohoto důvodu jsem program předělal tak, aby text svítil barvou, která je negativem barvy pozadí. Na obr. 6.6 je pro modré pozadí poměrně výrazně snížen jas, proto text svítí téměř bílou barvou s velmi slabým odstínem žluté. Zlepšení čitelnosti textu je více než patrné.



Obr. 6.6: Zobrazování textu se zapnutými pixely pro text

Nevýhodou tohoto řešení je podstatně větší odběr proudu, který tak musí být o to více softwarově omezen.

Závěr

Cílem této práce bylo navrhnout a následně realizovat hardwarové i programové řešení Chytré signalizační jednotky za účelem nahradit doposud nainstalované signalizační majáčky, a svými vlastnostmi zdokonalit stávající testbed samo-činného barmana.

Hned z úvodu za nadefinováním cílů a požadavků byl popsán vlastní význam signalizačních jednotek v průmyslu, a dále byly uvedeny jejich základní parametry pro výběr konkrétního produktu. Následoval průzkum dostupných signalizačních věží na trhu, v rámci kterého byl zjištěn např. fakt, že signální věže se vyrábějí maximálně s pěti moduly (patry).

Teoretickou část práce uvedla iniciativa Průmysl 4.0, na jejímž základě byl stručně představen testbed samo-činný barman. Dále bylo snahou popsat nejdůležitější oblasti, jimiž se práce zabývala.

Mezi zásadní požadavky na Chytrou signalizační jednotku patří především schopnost signalizovat stavy stroje podle průmyslového standardu PackML, který představuje obsáhlý soubor nejlepších praktik z oblasti automatizace shromážděných za dobu své dlouhé historie. V teoretické podkapitole zabývající se tímto standardem byl podrobně popsán základní stavový diagram obecné autonomní jednotky, již lze začlenit do většího systému. Následovala neméně důležitá sekce popisující význam jednotlivých signalizovaných barev.

Dalším požadavkem byla schopnost komunikovat přes průmyslový komunikační protokol Modbus TCP. Ten úzce navazuje na síťový protokol Ethernet, na kterém je postaven. Signalizační jednotka tak implementuje Modbus TCP server, který zajišťuje služby pro další připojená zařízení (Modbus TCP klienty). Podstatnou část práce tak představovala tvorba datového modelu, který definuje rozložení (mapování) adresového prostoru Modbus serveru a způsob zapisování do jeho registrů. S datovým modelem souvisela i nutnost nadefinovat datové typy, jež byly následně použity při softwarové implementaci firmwaru Chytré signalizační jednotky.

Velký důraz byl při návrhu koncepce, a následně při tvorbě desky plošných spojů, kladen také na síťovou technologii nazývanou Power over Ethernet (konkrétně na standard IEEE 802.3.at-2009 – PoE+). Díky ní mělo být dosaženo naprosté jednoduchosti při připojování zařízení k síti, jelikož by tak byl počet nutných kabelů zredukován na pouhý ethernetový kabel alespoň kategorie CAT5, po němž by bylo zařízení napájené a zároveň po něm komunikovalo. Vzhledem k odběru použitých digitálních LED pásků WS2812B byla tato technologie poměrně omezující z pohledu výkonu, který dokáže dodat. Ve finále toto nebyl až takový problém, jelikož použité LED pásky dostatečně svítily i po softwarovém omezení intenzity svitu.

Koncepce Chytré signalizační jednotky již spadá do praktické části práce. Ta se

zabývala způsoby zobrazování, které představují primární funkci signalizační jednotky. Nedílnou součástí je koncepce hardwaru, která popsala nejpodstatnější použité komponenty.

Závěrečná část práce se zabývala tvorbou desky plošných spojů ve vývojovém prostředí Eagle a realizací firmwaru pomocí softwarového nástroje Visual Studio Code za použití frameworku ESP-IDF od firmy Espressif.

Přestože deska plošných spojů byla navržena a vyrobena, nedošlo k finální realizaci z důvodu nedostatku několika čipů, které zastávaly důležitou roli v designu. Byl tak sestaven alespoň přípravek pro testování displeje složeného ze sedmi digitálních LED pásků, na němž byl laděn vytvořený firmware pro ovládání displeje. Zbytek programu, zajišťující komunikaci přes Ethernet a realizující Modbus TCP server, byl laděn na oficiální desce od výrobce Olimex, díky čemuž je zaručena skutečná funkčnost napsaného programu.

Během vývoje tohoto zařízení docházelo k několika zásadním změnám a především vize o využití technologie PoE+ s izolovanou zpětnou vazbou pomocí transformátoru, DC-DC převodníku, optočlenu a vysokonapěťového kondenzátoru podstatně prodražila celkový návrh Chytré signalizační jednotky. Z toho důvodu bude pravděpodobně od technologie PoE+ pro signalizační jednotku upuštěno. Práce ale může velmi dobře posloužit do budoucna jako zdroj informací při návrhu dalších zařízení podporujících technologii PoE.

Přesto ale přinesl tento návrh několik výhod oproti běžným signalizačním věžím. Těmi jsou především výrazně menší rozměry zařízení, možnost zobrazování až sedmi stavů naráz nebo využití textového módu.

Literatura

- [1] Schneider Electric. Signaling units Ø40-60-100: Harmony® XVC monolithic tower lights. [online]. 2013-9, s. 9. [cit. 2020-12-25]. Dostupné z: https://download.schneider-electric.com/files?p_enDocType=Catalog&p_File_Name=DIA5ED2130805EN.pdf
- [2] Wikipedia. Fourth Industrial Revolution. [online]. Poslední aktualizace 2020-12-21 [cit. 2020-12-21]. Dostupné z: https://en.wikipedia.org/wiki/Fourth_Industrial_Revolution
- [3] IoT Portál – redakce webu. CPS - 5C architektura. [online]. Zveřejněno 2016-08-22 [cit. 2020-12-22]. Dostupné z: <https://www.iot-portal.cz/2016/08/22/kyberfyzikalni-systemy/>
- [4] KING A. Industry 4.0 Design Principles. [online]. Zveřejněno [2018?] [cit. 2020-12-23]. Dostupné z: <https://www.rmit.edu.au/news/c4de/industry-4-0-design-principles>
- [5] Idexcel Technologies. Top 10 advantages of cloud computing. [online]. Zveřejněno 2017-10-17 [cit. 2020-12-22]. Dostupné z: <https://www.idexcel.com/blog/top-10-advantages-of-cloud-computing/>
- [6] KACZMARCZYK, V. – BAŠTÁN, O. – BRADÁČ, Z. – ARM, J. An Industry 4.0 Testbed (Self-Acting Barman): Principles and Design. *IFAC-PapersOnLine*. [online]. 2018, 51(6), s. 263–270 [cit. 2021-01-01]. ISSN 2405-8963. Dostupné z: <https://doi.org/10.1016/j.ifacol.2018.07.164>
- [7] Microsemi. Understanding 802.3at: PoE Plus Standard Increases Available Power. [online]. 2011. [cit. 2020-12-21]. Dostupné z: https://www.microsemi.com/document-portal/doc_download/24-understanding-ieee802-3at-poeplus
- [8] Silicon Labs. Understanding the IEEE 802.3bt PoE Standard. [online]. 2020? [cit. 2021-03-13]. Dostupné z: <https://www.silabs.com/documents/public/white-papers/understanding-the-ieee-8023bt-poe-standard.pdf>
- [9] Wikipedia. Power over Ethernet. [online]. Poslední aktualizace 2020-12-20 [cit. 2020-12-21]. Dostupné z: https://en.wikipedia.org/wiki/Power_over_Ethernet
- [10] Wikipedia. Modbus. [online]. Poslední aktualizace 2020-12-05 [cit. 2020-12-25]. Dostupné z: <https://en.wikipedia.org/wiki/Modbus>

- [11] Mostia, W. L. Introduction to Modbus. [online]. Zveřejněno 2019-01-04 [cit. 2020-12-25]. Dostupné z: <<https://www.controlglobal.com/articles/2019/introduction-to-modbus/>>
- [12] Modbus Organization. MODBUS Messaging on TCP/IP Implementation Guide V1.0b. [online]. 2006, s. 46. [cit. 2020-12-25]. Dostupné z: <https://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf>
- [13] Acromag, Inc. White Paper: Introduction To Modbus TCP/IP. [online]. 2012/2020, s. 10. [cit. 2021-04-09]. Dostupné z: <https://www.acromag.com/wp-content/uploads/2019/08/White-Paper-Introduction-to-ModbusTCP_765B-.pdf>
- [14] Modbus. Modbus Application Protocol Specification V1.1b3. [online]. 2012-04-26, s. 50. [cit. 2021-04-20]. Dostupné z: <https://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf>
- [15] Nøkleby, C. PackML unit/machine Implementation Guide: Part 1: PackML Interface State Manager. [online]. 2016-11-02, Ver. 1.00, s. 97. [cit. 2020-12-20]. Dostupné z: <http://omac.org/wp-content/uploads/2016/11/PackML_Unit_Machine_Implementation_Guide-V1-00.pdf>
- [16] Espressif Systems. ESP32 Series: Datasheet. [online]. 2020-04-27, Ver. 3.4, s. 57. [cit. 2020-12-28]. Dostupné z: <https://espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>
- [17] Espressif Systems. ESP32-WROOM-32: Datasheet. [online]. 2020-11, Ver. 3.0, s. 27. [cit. 2020-12-28]. Dostupné z: <https://espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf>
- [18] Microchip Technology Inc. LAN8710A/LAN8710Ai: Small Footprint MII/RMII 10/100 Ethernet Transceiver. [online]. 2016-07-15, Rev. B, s. 80. [cit. 2020-12-28]. Dostupné z: <<https://ww1.microchip.com/downloads/en/DeviceDoc/00002164B.pdf>>
- [19] Silicon Laboratories Inc. Si3406x Family Data Sheet. [online]. 2018-12, Rev. 1.0, s. 36. [cit. 2020-12-29]. Dostupné z: <<https://www.silabs.com/documents/public/data-sheets/si3406x-datasheet.pdf>>
- [20] ZEŽULKA, F. – HYNČICA, O. Průmyslový Ethernet III: Fyzické provedení sítě Ethernet. [online časopis]. *Automa*, 2007, roč. 13, č. 6, s. 40–44.

ISSN 1210-9592. [cit. 2020-12-27]. Dostupné z:
<[https://www.automa.cz/cz/casopis-clanky/
prumyslovy-ethernet-iii-fyzicke-provedeni-site-ethernet-2007_
06_34395_2402/](https://www.automa.cz/cz/casopis-clanky/prumyslovy-ethernet-iii-fyzicke-provedeni-site-ethernet-2007_06_34395_2402/)>

- [21] WCH. USB to serial chip CH340. [online]. 2018-07-09, Ver. 2C, s. 8. [cit. 2020-12-29]. Dostupné z:
<<https://cdn.sparkfun.com/assets/9/3/0/2/e/ch3402CDS.pdf>>
- [22] Maxim Integrated Products, Inc. DS18B20: Programmable Resolution 1-Wire Digital Thermometer. [online]. 2019/7, Ver. 6. [cit. 2021-04-09]. Dostupné z:
<<https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>>
- [23] Worldsemi. WS2812B: Intelligent control LED integrated light source. [online]. 2019-03-23, Ver. 5. [cit. 2021-04-09]. Dostupné z:
<<http://www.world-semi.com/DownloadFile/108>>

Seznam symbolů a zkratek

I 4.0	Industry 4.0
PoE	Power over Ethernet
PoE+	Power over Ethernet Plus
CPS	kyberneticko-fyzikální systém – Cyber-Physical System
IoT	Internet of Things
IoS	Internet of Services
24/7	nepřetržitě
3D	Three-Dimensional
ISO	International Organization for Standardization
OSI	Open Systems Interconnection
LAN	Local Area Network
DC	stejnoseměrný – Direct Current
PSE	Power-Sourcing Equipment
PD	Powered Device
MED	Media Endpoint Discovery
LLDP	Link Layer Discovery Protocol
MPS	Maintain Power Signature
OPC UA	Open Platform Communications Unified Architecture
MQTT	Message Queuing Telemetry Transport
PLC	Programmable Logic Controller
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
TCP	Transmission Control Protocol
IP	Internet Protocol

MAC	Media Access Control
PDU	Protocol Data Unit
ADU	Application Data Unit
MBAP	MODBUS Application Protocol
OMAC	Organization for Machine Automation and Control
ISA	International Society of Automation
AC	Alternating Current
SSU	Smart Signaling Unit
PackML	Packaging Machine Language
ID	Identifier
CPU	Central Processing Unit
BLE	Bluetooth Low Energy
Wi-Fi	Wireless Fidelity
SD	Secure Digital
UART	Universal Asynchronous Receiver/Transmitter
SPI	Serial Peripheral Interface
I^2C	Inter-Integrated Circuit
LED	Light-Emitting Diode
PWM	Pulse Width Modulation
I^2S	Inter-Integrated Circuit Sound
GPIO	General Purpose Input/Output
ADC	Analog-to-Digital Converter
DAC	Digital-to-Analog Converter
PHY	Physical Layer
MII	Media Independent Interface

RMII	Reduced Media Independent Interface
I/O	Input/Output
IC	Integrated Circuit
FET	Field Effect Transistor
DPS	deska plošných spojů
MPS	Maintain-Power-Signature
USB	Universal Serial Bus
FIFO	First In First Out
SC	State is Completed
HMI	Human-Machine Interface
DIN	Data Input
DOUT	Data Output
VDD	Voltage Drain Drain
SO	Small Outline
DO	Digital Output
GPI	General Purpose Input
EMI	Electromagnetic Induction
SMD	Surface Mount Device
ESP-IDF	Espressif IoT Development Framework
VSCoDe	Visual Studio Code
API	Application Programming Interface
UDP	User Datagram Protocol
TX	Transmit
RX	Receive

Veličiny	Značky	Jednotky	
výkon	P	W	(Watt)
napětí	U	V	(Volt)
proud	I	A	(Ampér)
rezistivita	R	Ω	(Ohm)
přenosová rychlost		b/s	(bitů za sekundu)
frekvence	f	Hz	(Hertz)
výkon na 1 miliwatt		dBm	(decibel-miliwatt)
relativní jednotka výkonu		dB	(decibel)

A Obsah elektronické přílohy

/	kořenový adresář přiloženého archivu
└─ A.1 Elektronická verze závěrečné práce..elektronická verze závěrečné práce	
└─ STEJSKAL-BP-Smart_Signaling_Unit.pdf	
└─ A.2 Schéma zapojení.....schéma zapojení SSU	
└─ Smart_Signaling_Unit.sch	
└─ A.3 Výrobní podklady DPS.....výrobní podklady pro výrobu DPS	
└─ Smart_Signaling_Unit.brd	
└─ A.4 Firmware.....firmware SSU	
└─ CMakeLists.txt.....hlavní konfigurační soubor pro CMake	
└─ sdkconfig.....konfigurační soubor projektu pro ESP-IDF	
└─ .vscode.....soubory pro nastavení Visual Studio Code a jeho rozšíření	
└─ c_cpp_properties.json	
└─ components.....komponenty CMake	
└─ display_driver	
└─ CMakeLists.txt	
└─ display_driver.cpp	
└─ include	
└─ display_driver.hpp	
└─ led_strip	
└─ led_strip_rmt_ws2812.c	
└─ include	
└─ led_strip.h	
└─ eth_comm	
└─ CMakeLists.txt	
└─ eth_comm.c	
└─ Kconfig.projbuild	
└─ include	
└─ eth_comm.h	
└─ mb_tcp_server	
└─ CMakeLists.txt	
└─ mb_tcp_server.cpp	
└─ include	
└─ mb_tcp_server.hpp	
└─ modbus_params	
└─ CMakeLists.txt	
└─ modbus_params.cpp	
└─ include	
└─ modbus_data_types.hpp	
└─ modbus_params.hpp	
└─ main	
└─ CMakeLists.txt	
└─ smart_signaling_unit.cpp.....hlavní spustitelný soubor	
└─ A.5 Datový model.....datový model SSU	
└─ Datový model.xlsx	